CXFS™ 7 Client-Only Guide for SGI® InfiniteStorage™

007–5619–010

New Features in this Guide

Note: Be sure to read the release notes to learn about any late-breaking changes to the installation and configuration procedures.

This revision contains clarifications to "Enable DMAPI for SLES 10 and SLES 11 Client-Only Nodes" on page 33.

Record of Revision

Version	Description
001	January 2010 Original publication with CXFS 6.0 and the SGI InfiniteStorage Software Platform (ISSP) release 2.0
002	September 2010 Supports the CXFS 6.2 product in the SGI ISSP 2.2 release
003	April 2011 Supports the CXFS 6.4 product in the ISSP 2.4 release.
004	April 2012 Supports the CXFS 6.6 product in the ISSP 2.6 release
005	April 2013 Supports the CXFS 7.0 product the ISSP 3.0 release
006	November 2013 Supports the CXFS 7.1 product the ISSP 3.1 release
007	May 2014 Supports the CXFS 7.2 product the ISSP 3.2 release
008	October 2014 Supports the CXFS 7.3 product the ISSP 3.3 release
009	July 2015 Supports the CXFS 7.4 product the ISSP 3.4 release
010	November 2015 Supports the CXFS 7.5 product the ISSP 3.5 release

Contents

Figures

Tables

About This Guide

This guide discusses the client-only platforms for the CXFS™ parallel-access filesystem for high-performance computing systems. For additional details, see the platform-specific release notes.

Prerequisites

This guide assumes the following:

- Server-capable administration nodes (running the supported operating system and CXFS software) are operational.
- The CXFS client-only nodes have the appropriate platform-specific operating system software installed.
- The reader is familiar with the information presented in the *CXFS 7 Administrator Guide for SGI InfiniteStorage* and the platform's operating system and installation documentation.

Related Publications

For information about this release, see the following release notes:

- SGI InfiniteStorage™ Software Platform (ISSP): `README.txt`
- CXFS:
  - `README_CXFS_GENERAL.txt`
  - `README_CXFS_LINUX.txt`
  - `README_CXFS_MACOSX.html`
  - `README_CXFS_WINDOWS.html`

The following documents contain additional information:

- CXFS documentation:
  - Platform-specific release notes
  - *CXFS 7 Administrator Guide for SGI InfiniteStorage*
- QLogic HBA card and driver documentation:

  http://www.qlogic.com
- Red Hat Linux documentation:

  http://www.redhat.com/docs/manuals/enterprise
- SUSE Linux Enterprise Software (SLES) documentation:

  http://www.suse.com/documentation
- Apple Mac OS X documentation:

  http://support.apple.com/manuals/#macos
- Microsoft Windows documentation:

  http://www.microsoft.com

Note: The external websites referred to in this guide were correct at the time of publication, but are subject to change.

The following man pages are provided on CXFS Linux and Mac OS X client-only nodes:

Client-Only Man Page	Linux RPM[1]
`cxfs_client`(8)	`cxfs_client`
`cxfs_info`(8)	`cxfs_client`
`cxfs-config`(8)	`cxfs_util`
`cxfscp`(1)	`cxfs_util`
`cxfsdump`(8)	`cxfs_util`

Obtaining Publications

You can obtain SGI documentation as follows:

- Log in to the SGI Customer Portal at http://support.sgi.com. Click the following:

  Support by Product
  > ***productname***
  > Documentation

  If you do not find what you are looking for, click **Search Knowledgebase**, enter a document-title keyword, select the category **Documentation**, and click **Search**

- On all but Windows systems, you can view man pages by typing `man` *title* at a command line.

- The `/docs` directory on the ISSP DVD or in the download directory contains the following:

  – The ISSP release note: `/docs/README.txt`

  – Other release notes: `/docs/README_`*NAME*`.txt`

1 For Mac OS X platforms, man pages are provided in the CXFS package.

  - A complete list of the packages and their location on the media: `/docs/RPMS.txt`
  - The packages and their respective licenses: `/docs/PACKAGE_LICENSES.txt`
- The release notes and manuals are provided in the `noarch/sgi-isspdocs` RPM and will be installed on the system into the following location:

  `/usr/share/doc/packages/sgi-issp-`*`ISSPVERSION-TITLE`*

Conventions

This guide uses the following terminology abbreviations:

- *Linux* refers to the supported Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES) as defined in the CXFS Linux release note
- *Mac OS X* refers to the supported releases as defined in the CXFS Mac OS X release note
- *Windows* refers to the supported levels of Microsoft Windows operating systems as defined in the CXFS Windows release note

The following conventions are used throughout this document:

Convention	Meaning
`command`	This fixed-space font denotes literal items such as commands, files, routines, path names, signals, messages, and programming language structures.
variable	Italic typeface denotes variable entries and words or concepts being defined.
`user input`	This bold, fixed-space font denotes literal items that the user enters in interactive sessions. (Output is shown in nonbold, fixed-space font.)
GUI	This font denotes the names of graphical user interface (GUI) elements such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, fields, and lists.
[ ]	Brackets enclose optional portions of a command or directive line.

...	Ellipses indicate that a preceding element can be repeated.
GUI element	This bold font denotes the names of graphical user interface (GUI) elements, such as windows, screens, dialog boxes, menus, toolbars, icons, buttons, boxes, and fields.
<TAB>	Represents pressing the specified key in an interactive session
`server-admin#`	In an example, this prompt indicates that the command is executed on a server-capable administration node
`client#`	In an example, this prompt indicates that the command is executed on a client-only node
`MDS#`	In an example, this prompt indicates that the command is executed on an active metadata server
#	In an example, this prompt indicates that the command is executed on an any node
specificnode#	In an example, this prompt indicates that the command is executed on a node named *specificnode* or of node type *specificnode*

Reader Comments

If you have comments about the technical accuracy, content, or organization of this publication, contact SGI. Be sure to include the title and document number of the publication with your comments. (Online, the document number is located in the front matter of the publication. In printed publications, the document number is located at the bottom of each page.)

You can contact SGI in either of the following ways:

- Send e-mail to the following address:

  techpubs@sgi.com

- Contact your customer service representative and ask that an incident be filed in the SGI incident tracking system:

  http://www.sgi.com/support/supportcenters.html

SGI values your comments and will respond to them promptly.

Introduction

This guide provide tells you how to install and configure clients for the CXFS™ parallel-access filesystem for high-performance computing environments. A *CXFS client-only node* has a minimal implementation of CXFS services that run a single daemon, the CXFS client daemon (`cxfs_client`). A cluster running multiple operating systems is known as a *multiOS cluster*.

For more information about CXFS terminology, concepts, and configuration, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Caution: CXFS is a complex product. To ensure that CXFS is installed and configured in an optimal manner, it is **mandatory** that you purchase SGI Installation Services developed for CXFS. Some of the procedures mentioned in this guide may be performed by SGI personnel or other qualified service personnel; details for these procedures are provided in other documents. Contact your local SGI sales representative for details.

This chapter discusses the following:

Also see Chapter 2, "Best Practices for Client-Only Nodes" on page 11.

CXFS on Client-Only Nodes

This section contains the following:

Client-Only Platforms

CXFS supports client-only nodes running any mixture of the following operating systems:

- Apple® Mac OS X®
- Red Hat® Enterprise Linux® (RHEL)
- SUSE® Linux® Enterprise Server (SLES)
- Microsoft® Windows®

For details, see the following:

- Chapter 3, "Linux Platform" on page 25
- Chapter 4, "Mac OS X Platform" on page 65
- Chapter 5, "Windows Platforms" on page 103

See the CXFS release notes for the supported kernels, update levels, and service pack levels.

Client-Only Commands

Table 1-1 lists the CXFS commands that are installed on all client-only nodes.

Table 1-1 CXFS Commands Available on All CXFS Client-Only Nodes

Command	Description
`cxfs_client`(8)	Controls the CXFS client control daemon
`cxfs_info`(8)	Provides status information
`cxfscp`(8)	Quickly copies large files (64 KB or larger) to and from a CXFS filesystem
`cxfsdump`(8)	Gathers configuration information in a CXFS cluster for diagnostic purposes
`grioadmin`(8)	Performs administrative tasks for the guaranteed-rate I/O product version 2 (GRIOv2)
`griomon`(8)	Monitors GRIO streams
`grioqos`(8)	Measures the quality-of-service metrics that GRIO maintains for each active stream
`xvm`(8)	Invokes the XVM command line interface

Also see:

- "CXFS Commands on Linux" on page 27
- "CXFS Commands on Mac OS X" on page 66
- "CXFS Commands on Windows" on page 105

Client-Only Installation and Configuration Overview

Following is the order of installation and configuration steps for a CXFS client-only node. See the *SGI InfiniteStorage Software Platform* release note and the specific operating system (OS) chapters in this guide for details:

1. Read the ISSP and CXFS release notes to learn about any late-breaking changes in the installation procedure.

2. Install the supported OS software according to the directions in the OS documentation.
3. Install and verify the RAID. See the *CXFS 7 Administrator Guide for SGI InfiniteStorage* and the release notes.
4. Install and verify the switch. See the *CXFS 7 Administrator Guide for SGI InfiniteStorage* and the release notes.
5. Obtain the supported CXFS server-side license keys. For more information about licensing, see "License Keys" on page 7 and *CXFS 7 Administrator Guide for SGI InfiniteStorage*.
6. Install and verify the host bus adapter (HBA) and driver.
7. Prepare the node, including adding a private network. See "Preinstallation Steps for Windows" on page 134.
8. Install the **SGI CXFS Clients** software onto one server-capable administration node and transfer the appropriate client packages to the corresponding client-only nodes, as described in the ISSP release note.
9. Perform any required post-installation configuration steps.
10. Configure the cluster to define the new client-only node, add it to the cluster, start CXFS services, and mount filesystems. See Chapter 6, "Configuring Client-Only Nodes" on page 197.

If you run into problems, see the OS-specific troubleshooting section, Chapter 7, "General Troubleshooting" on page 213, and the troubleshooting chapter in *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Cluster Administration

A CXFS cluster must contain at least one server-capable administration node that is responsible for updating that filesystem's metadata. This node is referred to as the *CXFS metadata server*. (Client-only nodes cannot be metadata servers.) Metadata servers store information in the CXFS cluster database. The CXFS cluster database is not stored on client-only nodes; only server-capable administration nodes contain the cluster database.

To modify the cluster database, you will use the CXFS graphical user interface (GUI) or the `cxfs_admin` command from a node with the correct permissions (usually a

server-capable administration node) and with `root` access. For more information about using these tools, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

CXFS Client Processes

When CXFS is started on a client-only node, a user-space daemon is started that provides the required processes. This is a subset of the processes needed on a CXFS server-capable administration node.

The `cxfs_client` daemon controls CXFS services on a client-only node. It does the following:

- Obtains the cluster configuration from a remote `fs2d` daemon and manages the local client-only node's CXFS kernel membership services and filesystems accordingly
- Obtains membership and filesystem status from the kernel

The path to the `cxfs_client` command varies among the platforms supported. See Appendix A, "Operating System Path Differences" on page 223.

Note: The `cxfs_client` daemon may still be running when CXFS services are disabled.

User Administration for CXFS

A CXFS cluster requires a consistent user identification scheme across all hosts in the cluster so that one person using different cluster nodes has the same access to the files on the cluster. The following must be observed to achieve this consistency:

- Users must have the same usernames on all nodes in the cluster. An individual user identifier (UID) should not be used by two different people anywhere in the cluster. Ideally, group names and group identifiers (GIDs) should also be consistent on all nodes in the cluster.
- Each CXFS client and server node must have access to the same UID and GID information. The simplest way to achieve this is to maintain the same `/etc/passwd` and `/etc/group` files on all CXFS nodes, but other mechanisms may be supported.

User and Group Quotas

Only Linux nodes can view or edit user and group quotas. Quotas are effective on all nodes because they are enforced by the metadata server.

To view or edit quota information on a Linux node, use the `xfs_quota` command. This is provided by the `xfsprogs` RPM.

Requirements

Using a CXFS client-only node requires the following:

- A supported storage area network (SAN) hardware configuration.

  Note: For details about supported hardware, see the Entitlement Sheet that accompanies the base CXFS release materials. (Using unsupported hardware constitutes a breach of the CXFS license.)

- A private 100baseT (or greater) TCP/IP network connected to each node, to be dedicated to the CXFS private heartbeat and control network. This network must not be a virtual local area network (VLAN) and the Ethernet switch must not connect to other networks. All nodes must be configured to use the same subnet.
- The appropriate license keys. See "License Keys" on page 7.
- A switch, which is required to protect data integrity on nodes without system controllers. See the release notes for supported switches.

  Nodes must use I/O fencing (or system reset if available) to protect the data integrity of the filesystems in the cluster. Server-capable administration nodes should use system reset. See "Protect Data Integrity" on page 14.
- There must be at least one server-capable administration node to act as the metadata server and from which to perform cluster administration tasks. You should install CXFS software on the server-capable administration nodes first.
- Nodes that are not potential metadata servers should be CXFS client-only nodes. A cluster may contain as many as 64 nodes, of which as many as 16 can be server-capable administration nodes; the rest must be client-only nodes. See "Make Most Nodes Client-Only Nodes" on page 14.

- Set the `mtcp_nodelay` system tunable parameter to `1` on server-capable administration nodes in order to provide adequate performance on file deletes.

Also see "Requirements for Windows" on page 104, and Chapter 2, "Best Practices for Client-Only Nodes" on page 11.

License Keys

CXFS requires the following license keys:

- CXFS license keys using server-side licensing. Server-side licensing is required on all nodes.

  To obtain server-side CXFS license keys, see information provided in your customer letter and the following web page:

  http://www.sgi.com/support/licensing

  The licensing used for server-capable administration nodes is based the SGI License Key (LK) software. See the general release notes and the *CXFS 7 Administrator Guide for SGI InfiniteStorage* for more information.

- Guaranteed rate I/O version 2 (GRIOv2) if enabled requires a license key on the server-capable administration nodes.

Guaranteed-Rate I/O (GRIO) and CXFS

CXFS supports guaranteed-rate I/O (GRIO) version 2 clients on all platforms, and GRIO servers on server-capable administration nodes. However, GRIO is disabled by default on server-capable administration nodes and Linux client-only nodes. See "GRIO on Linux" on page 52.

Note: GRIO application reservations are functional for Windows and Linux nodes; they are not functional on Mac OS X nodes.

After GRIO is enabled, the superuser can run the following commands from any node in the cluster:

- `grioadmin`, which provides stream and bandwidth management
- `grioqos`, which is the comprehensive stream quality-of-service monitoring tool

Run the above tools with the `-h` (help) option for a full description of all available options. See Appendix A, "Operating System Path Differences" on page 223, for the platform-specific locations of these tools.

See the platform-specific chapters in this guide for GRIO limitations and considerations:

- "GRIO on Linux" on page 52
- "GRIO on Mac OS X" on page 91
- "GRIO on Windows" on page 172

See the *Guaranteed-Rate I/O Version 2 for Linux Guide* for details about GRIO installation, configuration, and use.

XVM Failover and CXFS

XVM failover version 2 (v2) requires that the RAID be configured in AVT mode.

To configure failover v2, you must create and edit the `failover2.conf` file. For more information, see the following:

- The comments in the `failover2.conf` file on a server-capable administration node
- *CXFS 7 Administrator Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator Guide*

This guide contains platform-specific examples of `failover2.conf` for the following:

- "Configuring XVM Failover V2 on Linux" on page 44
- "Configuring XVM Failover V2 on Mac OS X" on page 86
- "Configuring the `failover2.conf` File for Windows" on page 146

Monitoring CXFS

To monitor CXFS, you can use the following:

- The `cxfs_info` command on the client
- The view area of the CXFS GUI
- The `cxfs_admin` command
- The `clconf_info` command on a CXFS server-capable administration node

For more information, see "Verifying the Cluster Status" on page 205.

Chapter 2

Best Practices for Client-Only Nodes

This chapter discusses best-practices for client-only nodes:

Also see the best practices information in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Configuration Best Practices

This section discusses the following:

Understand Hostname Resolution and Network Configuration Rules

Caution: It is critical that you understand these rules before attempting to configure a CXFS cluster.

The following hostname resolution rules and recommendations apply to all nodes:

- You must ensure that the hostname and IP address for each network interface in the cluster is properly configured on each client-only node and server-capable administration node.
- The first node you define must be a server-capable administration node.
- Hostnames cannot begin with an underscore (_) or include any whitespace characters.
- The private network IP addresses on a running node in the cluster cannot be changed while CXFS services are active.
- You must be able to communicate directly between every node in the cluster (including client-only nodes) using IP addresses and logical names, without routing.
- A private network must be dedicated to be the heartbeat and control network. No other load is supported on this network.
- The heartbeat and control network must be connected to all nodes, and all nodes must be configured to use the same subnet for that network.

If you change hostname resolution settings in the `/etc/nsswitch.conf` file after you have defined the first server-capable administration node (which creates the cluster database), you must recreate the cluster database.

To confirm network connectivity, use the following command line on a server-capable administration node:

```
server-admin# /usr/cluster/bin/cxfs-config -check -ping
```

For more information, see *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Fix Network Issues First

If there are any network issues on the private network, fix them before trying to use CXFS. A stable private network is important for a stable CXFS cluster network. Ensure that you understand the information in "Understand Hostname Resolution and Network Configuration Rules" on page 12.

When you install the CXFS software on the client-only node, you must modify certain system files. **The network configuration is critical.** Each node in the cluster must be able to communicate with every other node in the cluster by both logical name and IP address without going through any other network routing; proper name resolution is key. SGI recommends static routing.

Use a Private Network

You are required to use a private network for CXFS metadata traffic:

- The private network is used for metadata traffic and should not be used for other kinds of traffic.
- A stable private network is important for a stable CXFS cluster environment.
- Two or more clusters should not share the same private network. A separate private network switch is required for each cluster.
- The private network should contain at least a 100-Mbit network switch. A network hub is not supported and should not be used.
- All cluster nodes should be on the same physical network segment (that is, no routers between hosts and the switch).
- Use private (10.*x.x.x*, 176.16.*x.x*, or 192.168.*x.x*) network addresses (RFC 1918).
- The private network must be configured as the highest priority network for the cluster. The public network may be configured as a lower priority network to be used by CXFS network failover in case of a failure in the private network.
- When administering more than one CXFS cluster, use unique private network addresses for each cluster. If you have multiple clusters connected to the same public network, use unique cluster names and cluster IDs.
- A virtual local area network (VLAN) is not supported for a private network.

- When NFS or Samba serving from a CXFS cluster, the network used for remote fileserving cannot be a backup private network for CXFS. Using the fileserving network as a backup private network for CXFS private network may result in heartbeat timeouts, which will cause a severe drop in CXFS and fileserving performance.

Make Most Nodes Client-Only Nodes

You should define most nodes as client-only nodes and define just the nodes that may be used for CXFS metadata as server-capable administration nodes.

The advantage to using client-only nodes is that they do not keep a copy of the cluster database; they contact a server-capable administration node to get configuration information. It is easier and faster to keep the database synchronized on a small set of nodes, rather than on every node in the cluster. In addition, if there are issues, there will be a smaller set of nodes on which you must look for problems.

Use the Correct Mix of Software Releases

All nodes should run the same level of CXFS and the same level of operating system software, according to platform type. To support upgrading without having to take the whole cluster down, nodes can run different CXFS releases during the upgrade process.

Caution: You must upgrade all server-capable administration nodes before upgrading any client-only nodes (servers must run the same release as client-only nodes or a later release.) Operating a cluster with clients running a mixture of CXFS versions will result in a performance loss. Relocation to a server-capable administration node that is running an older CXFS version is not supported.

For details, see the platform-specific release notes and the information about rolling upgrades in *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Protect Data Integrity

All nodes must be configured to protect data integrity in case of failure. System reset or I/O fencing is required to ensure data integrity for all nodes. I/O fencing (or system reset when available) must be used on client-only nodes.

You should use the `admin` account when configuring I/O fencing. You must limit the switch to a single login session for the `admin` account. For details, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

You must keep the `telnet` or `ssh` (Brocade only) port on the switch free at all times; **do not** leave multiple login sessions connected.

SGI recommends that you use a switched network of at least 100baseT.

You should isolate the power supply for the switch from the power supply for a node and its system controller. You should avoid any possible situation in which a node can continue running while both the switch and the system controller lose power. Avoiding this situation will prevent the possibility a split-brain scenario.

You must put switches used for I/O fencing on a network other than the primary CXFS private network so that problems on the CXFS private network can be dealt with by the fencing process and thereby avoid data corruption issues. The network to which the switch is connected must be accessible by all server-capable administration nodes in the cluster.

See the following:

- "Configuring I/O Fencing for Linux" on page 45
- "Configuring I/O Fencing for Mac OS X" on page 88
- "Configuring I/O Fencing for Windows (FC Only)" on page 163

Use a Client-Only Tiebreaker

SGI recommends that you always define a stable client-only node as the CXFS tiebreaker for all clusters with more than one server-capable administration node and at least one client-only node.

Having a tiebreaker is critical when there are an even number of server-capable administration nodes. A tiebreaker avoids the problem of multiple-clusters being formed (a split cluster) while still allowing the cluster to continue if one of the metadata servers fails.

As long as there is a reliable client-only node in the cluster, a client-only node should be used as tiebreaker. Server-capable administration nodes are not recommended as tiebreaker nodes because these nodes always affect CXFS kernel membership.

The tiebreaker is of benefit in a cluster even with an odd number of server-capable administration nodes because when one of the server-capable administration nodes is removed from the cluster, it effectively becomes a cluster with an even-number of server-capable administration nodes.

Note the following:

- If exactly two server-capable administration nodes are configured and there are no client-only nodes, **neither** server-capable administration node should be set as the tiebreaker. (If one node was set as the tiebreaker and it failed, the other node would also shut down.)
- If exactly two server-capable administration nodes are configured and there is at least one client-only node, you should specify the client-only node as a tiebreaker.

  If one of the server-capable administration nodes is the CXFS tiebreaker in a two-server-capable-node cluster, failure of that node or stopping the CXFS services on that node will result in a cluster-wide forced shutdown. If you use a client-only node as the tiebreaker, either server-capable administration node could fail but the cluster would remain operational via the other server-capable administration node.
- If there are an even number of server-capable administration nodes and there is no tiebreaker set, the fail policy must not contain the `shutdown` option because there is no notification that a shutdown has occurred.

SGI recommends that you start CXFS services on the tiebreaker client after the server-capable administration nodes are all up and running, but before CXFS services are started on any other clients.

Enable Forced Unmount When Appropriate

Normally, an unmount operation will fail if any process has an open file on the filesystem. The *forced unmount* feature allows the unmount to proceed regardless of whether the filesystem is still in use.

If you enable the forced unmount feature for CXFS filesystems (which is turned off by default), you may be able to improve the stability of the CXFS cluster, particularly in situations where the filesystem must be unmounted. However, be aware that a forced unmount will kill running processes to unmount a filesystem, which is potentially destructive.

For more information, see "Forcing Unmount of CXFS Filesystems" on page 203 and the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Configure Firewalls for CXFS Use

Do one of the following:

- Configure firewalls to allow CXFS traffic. See *CXFS 7 Administrator Guide for SGI InfiniteStorage* for CXFS port usage. (Preferred for most platforms.)
- Configure firewalls to allow all traffic on the CXFS private interfaces. This assumes that the public interface is not a backup metadata network.
- Disable firewalls. (Preferred for Windows. For Windows Vista® and Windows 2008, you should check firewall settings after each reboot.)

For more information, see your firewall documentation.

Use the LSI Drivers that Ship with the Linux OS

You should use the LSI drivers that ship with the Linux operating systems. The newer drivers that are available on the LSI web site may not work and are not supported by SGI or the kernels CXFS uses in this release.

Administration Best Practices

This section discusses the following:

Upgrade the Software Properly

Do the following when upgrading the software:

- Save the current CXFS configuration as a precaution before you start an upgrade and acquire new CXFS server-side licenses (if required). See the information in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.
- Read the release notes and any late-breaking caveats on the download page before installing or upgrading CXFS. These notes contain useful information and caveats needed for a stable install/upgrade.
- Do not make any other configuration changes to the cluster (such as adding new nodes or filesystems) until the upgrade of all nodes is complete and the cluster is running normally.

See the following:

Understand the Platform-Specific Limitations and Considerations

Each platform in a CXFS cluster has different issues. See the following:

- "Limitations and Considerations for Linux" on page 30
- "Limitations and Considerations for Mac OS X" on page 69
- "Functional Limitations and Considerations for Windows" on page 111 and "Performance Considerations for Windows" on page 119

Shut Down Client-Only Nodes Properly

When shutting down, resetting, or restarting a CXFS client-only node, do not stop CXFS services on the node. (Stopping CXFS services is more intrusive on other nodes in the cluster because it updates the cluster database. Stopping CXFS services is appropriate only for a CXFS server-capable administration node.) Rather, let the CXFS shutdown scripts on the node stop CXFS when the client-only node is shut down or restarted.

Do Not Run Backups on a Client Node

SGI recommends that you perform backups on the CXFS metadata server.

Do not run backups on a client node, because it causes heavy use of non-swappable kernel memory on the metadata server. During a backup, every inode on the filesystem is visited; if done from a client, it imposes a huge load on the metadata server. The metadata server may experience typical out-of-memory symptoms, and in the worst case can even become unresponsive or crash.

Use `cron` Jobs Properly

Jobs scheduled with `cron` can cause severe stress on a CXFS filesystem if multiple nodes in a cluster start the same filesystem-intensive task simultaneously.

Because CXFS filesystems are considered as local on all nodes in the cluster, the nodes may generate excessive filesystem activity if they try to access the same filesystems simultaneously while running commands such as `find` or `ls`. You should build databases for `rfind` and GNU `locate` only on the active metadata server.

Any task initiated using `cron` on a CXFS filesystem should be launched from a single node in the cluster, preferably from the active metadata server. Edit the nodes' `crontab` file to only execute the `find` command on one metadata server of the cluster.

Repair Filesystems with Care

Always contact SGI technical support before using `xfs_repair` on CXFS filesystems. You must first ensure that you have an actual case of filesystem corruption and retain valuable metadata information by replaying the XFS logs before running `xfs_repair`.

Caution: If you run `xfs_repair` without first replaying the XFS logs, you may introduce data corruption. You should run `xfs_ncheck` and capture the output to a file before running `xfs_repair`. If running `xfs_repair` results in files being placed in the `lost+found` directory, the saved output from `xfs_ncheck` may help you to identify the original names of the files.

Only use `xfs_repair` on server-capable administration nodes and only when you have verified that all other cluster nodes have unmounted the filesystem. Make sure that `xfs_repair` is run only on a cleanly unmounted filesystem. If your filesystem has not been cleanly unmounted, there will be uncommitted metadata transactions in the log, which `xfs_repair` will erase. This usually causes loss of some data and messages from `xfs_repair` that make the filesystem appear to be corrupted.

If you are running `xfs_repair` right after a system crash or a filesystem shutdown, your filesystem is likely to have a dirty log. To avoid data loss, you **MUST** mount and unmount the filesystem before running `xfs_repair`. It does not hurt anything to mount and unmount the filesystem locally, after CXFS has unmounted it, before `xfs_repair` is run.

Disable CXFS Before Maintenance

You should disable CXFS before maintenance as follows:

1. Perform a forced CXFS shutdown.
2. Stop the `cxfs_client` daemon.
3. Disable `cxfs_client` from automatically restarting.

Do Not Run Power Management Software

Do not run power management software, which may interfere with the CXFS cluster.

Use Fast Copying for Large CXFS Files

You can use the `cxfscp`(1) command to quickly copy large files (64 KB or larger) to and from a CXFS filesystem. It can be significantly faster than `cp`(1) on CXFS filesystems because it uses multiple threads and large direct I/Os to fully use the bandwidth to the storage hardware.

Files smaller than 64 KB do not benefit from large direct I/Os. For these files, `cxfscp` uses a separate thread using buffered I/O, similar to `cp`(1).

The `cxfscp` command is available on Linux and Windows platforms. However, some options are platform-specific, and other limitations apply. For more information and a complete list of options, see the `cxfscp`(1) man page.

Appropriately Map Physical Device Names to XVM Physvols

To match up physical device names to their corresponding XVM physical volumes (*physvols*), use the following command:

```
# xvm show -v -top -ext vol/volname
```

In the output for this command, the information within the parentheses matches up the XVM pieces with the device name. For example (line breaks shown for readability):

```
# xvm show -v -top -ext vol/test
vol/test                            0 online,open
    subvol/test/data         1142792192 online,open
        stripe/stripe0           1142792192 online,tempname,open (unit size:128)
            slice/cc_is4500-lun0-gpts0  142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
            slice/cc_is4500-lun1-gpts0  142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
            slice/cc_is4500-lun0-gpts1  142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
            slice/cc_is4500-lun1-gpts1  142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
            slice/cc_is4500-lun0-gpts2  142849024 online,open
```

```
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
            slice/cc_is4500-lun1-gpts2  142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
            slice/cc_is4500-lun0-gpts3  142849024 online,open
(cc_is4500-lun0-gpt:/dev/xscsi/pci08.03.0/node200400a0b8119204/port4/lun0/disc)
            slice/cc_is4500-lun1-gpts3  142849024 online,open
(cc_is4500-lun1-gpt:/dev/xscsi/pci08.03.1/node200500a0b8119204/port1/lun1/disc)
```

Note: The `xvm` command on the Windows platform does not display the worldwide name (WWN). For more information about WWNs and Windows, see "Configuring the `failover2.conf` File for Windows" on page 146.

For more information about XVM physvols, see the *XVM Volume Manager Administrator Guide*.

Do Not Overfill CXFS Filesystems

For best performance, keep your CXFS filesystems under 98% full. This is also a best practice for a local filesystem, but is even more important for a CXFS filesystem because of fragmented files and increased metadata traffic.

Limit Client Accounts to 32 Groups

The CXFS metadata server is only capable of managing permissions for users with 32 or fewer group memberships. Therefore, all accounts (including `root`) on CXFS clients must be limited to 32 or fewer groups.

Turn Off Logical Volume Manager on Linux Nodes if Unused

If you do not have a local XVM volume on your Linux system, you should disable the Logical Volume Manager to avoid unnecessarily probing all of the disks and `lun0` LUNs to which the machine has access. Do the following:

- RHEL 7 or SLES 12:

```
rhel7_or_sles12# systemctl mask lvm2-activation-early.service
rhel7_or_sles12# systemctl mask lvm2-activation.service
rhel7_or_sles12# systemctl mask lvm2-lvmetad.service
rhel7_or_sles12# systemctl mask lvm2-lvmetad.socket
rhel7_or_sles12# systemctl mask lvm2-monitor.service
```

- Earlier Linux:

```
earlierlinux# chkconfig boot.lvm off
```

Access the Correct Cluster at a Multiple-Cluster Site

If you have multiple clusters connected to the same public network, you should add `-i` *clustername* to the `cxfs_client.options` file.

Note: CXFS does not support multiple clusters on the same **private** network.

Use Consistent Kernel System Tunable Parameter Settings

SGI recommends that you use the same settings on kernel system tunable-parameters on all applicable nodes in the cluster. You should only modify the parameters if advised to do so by SGI Support.

The system tunable parameters vary by client OS. For more information, see:

- "System Tunable Kernel Parameters on Linux" on page 53
- "System Tunable Kernel Parameters on Mac OS X" on page 91
- "System-Tunable Parameters for Windows" on page 173
- The appendix about system tunable parameters in *CXFS 7 Administrator Guide for SGI InfiniteStorage*

Linux Platform

CXFS supports a client-only node running the Red Hat Enterprise Linux (RHEL) or SUSE Linux Enterprise Server (SLES) operating system, as defined in the CXFS Linux release notes.

Note: Nodes that you intend to run as metadata servers must be installed as server-capable administration nodes; all other nodes should be client-only nodes. For information about server-capable administration nodes, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

This chapter contains the following sections:

CXFS on Linux

This section contains the following information about CXFS on Linux systems:

Requirements for Linux

In addition to the items listed in "Requirements" on page 6, using a Linux node to support CXFS requires the following, as detailed in the CXFS Linux release note:

- One of the following distributions:
  - RHEL
  - SLES

  See the release notes for the supported kernels, update levels, and service pack levels.
- Supported Fibre Channel, serial-attached storage (SAS), or InfiniBand switches. Either system reset or I/O fencing is required for all nodes.

  Note: See Appendix E, "L2 System Controller for Linux Reset" on page 237.
- A choice of at least one supported SAN host bus adapter (HBA)
- A CPU of the following class:

– x86_64 architecture, such as:

  • AMD Opteron

  • Intel Xeon EM64T

– ia64 architecture, such as Intel Itanium 2

The machine must have at least the following **minimum** requirements:

– 256 MB of RAM memory

– Two Ethernet 100baseT interfaces

– One empty PCI slot (to receive the HBA)

CXFS Commands on Linux

The following commands are shipped as part of the CXFS Linux package:

```
/usr/cluster/bin/cxfs_admin
/usr/cluster/bin/cxfs_client
/usr/cluster/bin/cxfs_info
/usr/cluster/bin/cxfscp
/usr/cluster/bin/cxfsdump
/usr/cluster/bin/framesort
/usr/cluster/bin/frametest
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/grioqos
/sbin/xvm
```

For more information about these commands, see the man pages and the *CXFS 7 Administrator Guide for SGI InfiniteStorage*

Note the following:

- The `cxfs_client` and `xvm` commands are needed to include a client-only node in a CXFS cluster.
- The `cxfs_info` command reports the current status of this node in the CXFS cluster.
- The `rpm` command output lists all software added; see "Installing CXFS Software for Linux" on page 40.

- To make administrative changes via `cxfs_admin` from a client-only node, you must first use the `cxfs_admin access` command on a server-capable administration node to grant `admin` permission to the client-only node. For more information, see the section about setting `cxfs_admin` access permissions in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Log Files on Linux

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. You should monitor the `/var/log/cxfs_client` and `/var/log/messages` log files for problems. Look for a `Membership delivered` message to indicate that a cluster was formed.

The Linux platform uses the `logrotate` system utility to rotate the CXFS logs (as opposed to other multiOS platforms, which use the `-z` option to `cxfs_client`):

- The `/etc/logrotate.conf` file specifies how often system logs are rotated
- The `/etc/logrotate.d/cxfs_client` file specifies the manner in which `cxfs_client` logs are rotated

For information about the log files created on server-capable administration nodes, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

CXFS Scripts on Linux

The following CXFS scripts are provided for execution by the `cxfs_client` daemon:

- "Mount Scripts" on page 28
- "`cxfs-reprobe` Script" on page 29
- "`cxfs-enumerate-wwns` Script" on page 29

Mount Scripts

The `cxfs_client` executes the CXFS mount scripts before a CXFS filesystem is mounted and after a CXFS filesystem is unmounted on a Linux client-only node. You can customize these scripts to suit a particular environment. For example, an application could be started when a CXFS filesystem is mounted by extending the

`cxfs-post-mount` script. The application could be terminated by changing the `cxfs-pre-umount` script. The mount scripts are installed in the following locations:

```
/var/cluster/cxfs_client-scripts/cxfs-pre-mount
/var/cluster/cxfs_client-scripts/cxfs-post-mount
/var/cluster/cxfs_client-scripts/cxfs-pre-umount
/var/cluster/cxfs_client-scripts/cxfs-post-umount
```

For more details about using these scripts, and for information about the mount scripts on server-capable administration nodes, see *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

`cxfs-reprobe` Script

CXFS uses the `cxfs-reprobe` script to ensure that LUN path failover works after fencing. The `cxfs_client` daemon runs `cxfs-reprobe` to reprobe the storage controllers on client-only nodes when they join or rejoin membership. The script is installed in the following location:

```
/var/cluster/cxfs_client-scripts/cxfs-reprobe
```

Note: In order for `cxfs-reprobe` to appropriately probe all of the targets on the SCSI bus, you must define a group of environment variables in the `/etc/cluster/config/cxfs_client.options` file.

`cxfs-enumerate-wwns` Script

The `cxfs-enumerate-wwns` script enumerates the host's world wide names (WWNs) that are known to CXFS. For example:

```
linux# /var/cluster/cxfs_client-scripts/cxfs-enumerate-wwns
# cxfs-enumerate-wwns
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# lsi @ /proc/mpt/ioc2/info
100000062b0f4ff1
```

```
# lsi @ /proc/mpt/ioc1/info
100000062b0f4ff0
# lsi @ /proc/mpt/ioc0/info
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host5
100000062b0f4ff0
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
# fc_host @ /sys/class/fc_host/host6
100000062b0f4ff1
```

Limitations and Considerations for Linux

Note the following:

- On Linux systems, the use of XVM is supported only with CXFS; XVM does not support local Linux disk volumes.
- On systems running SLES 10 that are greater than 64 CPUs, there are issues with using the `md` driver and CXFS. The `md` driver holds the BKL (Big Kernel Lock), which is a single, system-wide spin lock. Attempting to acquire this lock can add substantial latency to a driver's operation, which in turn holds off other processes such as CXFS. The delay causes CXFS to lose membership. This problem has been observed specifically when an md pair RAID split is done, such as the following:

  ```
  raidsetfaulty /dev/md1 /dev/path/to/partition
  ```

- Although it is possible to mount other filesystems on top of a Linux CXFS filesystem, this is not recommended.

- CXFS filesystems with XFS version 1 directory format cannot be mounted on Linux nodes.
- The implementation of file creates using `O_EXCL` is not complete. Multiple applications running on the same node using `O_EXCL` creates as a synchronization mechanism will see the expected behavior (only one of the creates will succeed). However, applications running between nodes may not get the `O_EXCL` behavior they requested (creates of the same file from two or more separate nodes may all succeed).
- The SAN HBA driver must be loaded before CXFS services are started. The HBA driver could be loaded early in the initialization scripts or be added to the initial RAM disk for the kernel. See the `mkinitrd` man page for more information.
- RHEL 5 x86_64 nodes have a severely limited kernel stack size. To use CXFS on these nodes requires the following to avoid a stack overflow panic:
  - You must fully disable SELinux on x86_64 RHEL 5 client-only nodes (you cannot simply set it to `permissive` mode). For more information, see:

    http://www.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/5.2/html/Deployment_Guide/sec-sel-enable-disable.html

    Note: This caveat does not apply to RHEL 5 nodes with ia64 architectures.

- Case-insensitive CXFS filesystems are not supported on RHEL 4, RHEL 5, and SLES 10 client-only nodes. These nodes will fail to mount the filesystem with messages such as the following:

```
Preparing to mount CXFS file system "/dev/cxvm/tp91"
XFS: bad version
XFS: SB validate failed
```

- An export option called `no_sendfile` has been added to the enhanced NFS server for SLES 10 systems. If you are having an issue with a SLES 10 client serving NFS, SGI recommendeds that you use `no_sendfile`. For more information, see the `exports`(5) man page.
- Filesystems created with the default `mkfs` parameters will not mount on RHEL 4 U3 systems because they do not support filesystems with `attr=2`.

- Older filesystems created under IRIX with directory-naming suboption `version=1` cannot be mounted on Linux.
- RHEL 4 and RHEL 5 clients cannot mount filesystems built with `lazy-count=1`. For RHEL clients, you must build the filesystems with the following required options:
  - RHEL 4 (any update):

    ```
    server# mkfs -t xfs -l lazy-count=0 -i attr=1
    ```

  - RHEL 5 (any update):

    ```
    server# mkfs -t xfs -l lazy-count=0
    ```

  Depending upon version of the `mkfs.xfs` program that is installed, these options may or may not be the default. The parameters used are printed by `mkfs.xfs` when run, and can also be obtained later by using the following command on SLES systems:

  ```
  sles# xfs_info mountpoint
  ```

- If you are installing the CXFS client package on a system that is currently running XVM from the XVM Standalone for SLES distribution, you may see messages similar to the following:

```
WARNING: /lib/modules/2.6.16.21-0.8-smp/weak-updates/xvm/sgi-xvm-cell.ko needs unknown symbol xvm_trace_enter
WARNING: /lib/modules/2.6.16.21-0.8-smp/weak-updates/xvm/sgi-xvm-cell.ko needs unknown symbol xvm_physlab_ver_to_cur
```

  You should ignore these messages and reboot the system as documented in the installation instructions.

- On Linux client-only nodes, memory-mapping an offline file in a DMF filesystem may cause other processes such as `ps`(1) to block while DMF is making the file online.
- On a RHEL client-only node, if an application uses a memory-mapped file within a DMF filesystem and DMF subsequently makes the file offline, the application could see zeros instead of its data for subsequent pages brought into memory as the result of page faults. Similarly, if a file in a DMF filesystem is memory-mapped and then changed, it is possible for those changes to be lost if DMF subsequently makes the file offline.
- CXFS does not support the enhanced XFS `agskip` mount option on RHEL 4 or RHEL 5 clients.

See also:

- Appendix B, "Filesystem and Logical Unit Specifications" on page 227
- The appendix about `mkfs` options and CXFS in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*

Enable DMAPI for SLES 10 and SLES 11 Client-Only Nodes

Note: This section does not apply to SLES 12. On SLES 12 systems, DMAPI is enabled by default for CXFS clients.

You must enable DMAPI for SLES 10 and SLES 11 client-only nodes that are not running DMF software if you want to mount DMF-managed CXFS filesystems. Use the `dmi` mount option and set `DMAPI_PROBE="yes"` in the `/etc/sysconfig/sysctl` file on the node. Changes to the file will processed on the next reboot.

After setting that system configuration file, you can immediately enable DMAPI for the current boot session by executing the following:

```
# sysctl -w fs.xfs.probe_dmapi=1
```

Note: For those nodes that are also DMF servers or DMF parallel data-mover nodes, DMAPI is automatically enabled when installing the `dmf` or `dmf-mover` packages.

Access Control Lists and Linux

All CXFS files have UNIX mode bits (read, write, and execute) and optionally an access control list (ACL). For more information about POSIX ACLs, see the `chmod` and `setfacl` man pages.

Fibre Channel HBA Installation for Linux

This section provides an overview of the Fibre Channel HBA installation information for Linux nodes.

The installation may be performed by you or by a qualified service representative for your hardware. See the Linux operating system documentation and the documentation for your hardware platform.

The driver requirements are as follows:

- LSI Logic card: the drivers are supplied with the Linux kernel. The module names are `mptscsih` and `mptfc`. The LSI `lsiutil` command displays the number of LSI HBAs installed, the model numbers, and firmware versions.
- QLogic card: the drivers are supplied with the Linux kernel.

You must ensure that the HBA driver is loaded prior to CXFS initialization by building the module into the initial RAM disk automatically or manually.

For example, using the QLogic card and the `qla2200` driver:

- **Automatic method:**

  Add a new line such as the following to the `/etc/modprobe.d/sgi-cxfs-xvm.conf` file:

  ```
  alias scsi_hostadapter1 qla2200
  ```

  Note: You may have to create this file when adding the first parameter.

  For SLES, add the driver name to the `INITRD_MODULES` variable in the `/etc/sysconfig/kernel` file. After adding the HBA driver into `INITRD_MODULES`, you must rebuild `initrd` with `mkinitrd`.

  Note: If the host adapter is installed in the box when the operating system is installed, this may not be necessary. Or hardware may be detected at boot time.

  When the new kernel is installed, the driver will be automatically included in the corresponding `initrd` image.

- **Manual method:**

  Recreate your `initrd` to include the appropriate HBA driver module. For more information, see the operating system documentation for the `mkinitrd` command.

You should then verify the appropriate `initrd` information:

- If using the GRUB loader, verify that the following line appears in the `/boot/grub/grub.conf` file:

  `initrd /initrd-`*version*`.img`

- If using the LILO loader, do the following:

  1. Verify that the following line appears in the appropriate stanza of `/etc/lilo.conf`:

     `/boot/initrd-`*version*`.img`

  2. Rerun LILO.

The system must be rebooted (and when using LILO, LILO must be rerun) for the new `initrd` image to take effect.

Instead of this procedure, you could also modify the `/etc/rc.sysinit` script to load the `qla2200` driver early in the `initscript` sequence.

Preinstallation Steps for Linux

This section provides an overview of the steps that you will perform on your Linux nodes prior to installing the CXFS software. It contains the following sections:

- "Adding a Private Network for Linux" on page 35
- "Using CXFS GUI Connectivity Diagnostics for Linux" on page 38
- "Verifying the Private and Public Networks for Linux" on page 39

Adding a Private Network for Linux

The following procedure provides an overview of the steps required to add a private network to the Linux system. A private network is required for use with CXFS. See "Use a Private Network" on page 13.

You may skip some steps, depending upon the starting conditions at your site. For details about any of these steps, see the Linux operating system documentation.

1. Edit the `/etc/hosts` file so that it contains entries for every node in the cluster and their private interfaces as well. The `/etc/hosts` file has the following format, where *primary_hostname* can be the simple hostname or the fully qualified domain name:

   IP_address primary_hostname aliases

   You should be consistent when using fully qualified domain names in the `/etc/hosts` file. If you use fully qualified domain names on a particular node, then all of the nodes in the cluster should use the fully qualified name of that node when defining the IP/hostname information for that node in their `/etc/hosts` file.

   The decision to use fully qualified domain names is usually a matter of how the clients (such as NFS) are going to resolve names for their client server programs, how their default resolution is done, and so on.

   Even if you are using the domain name service (DNS) or the network information service (NIS), you must add every IP address and hostname for the nodes to `/etc/hosts` on all nodes. For example:

   ```
   190.0.2.1 server1.company.com server1
   190.0.2.3 stocks
   190.0.3.1 priv-server1
   190.0.2.2 server2.company.com server2
   190.0.2.4 bonds
   190.0.3.2 priv-server2
   ```

   You should then add all of these IP addresses to `/etc/hosts` on the other nodes in the cluster.

   For more information, see the `hosts` and `resolver` man pages.

   Note: Exclusive use of NIS or DNS for IP address lookup for the nodes will reduce availability in situations where the NIS or DNS service becomes unreliable.

   For more information, see "Understand Hostname Resolution and Network Configuration Rules" on page 12.

2. Edit the `/etc/nsswitch.conf` file so that local files are accessed before either NIS or DNS. That is, the hosts line in `/etc/nsswitch.conf` must list `files` first. For example:

   ```
   hosts:      files nis dns
   ```

   (The order of `nis` and `dns` is not significant to CXFS, but `files` must be first.)

3. Configure your private interface according to the instructions in the network configuration section of your Linux distribution manual. To verify that the private interface is operational, issue the following command:

   ```
   linux# ifconfig -a
   ```

   For example:

```
linux# ifconfig -a

eth0      Link encap:Ethernet  HWaddr 00:50:81:A4:75:6A
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13782788 errors:0 dropped:0 overruns:0 frame:0
          TX packets:60846 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:826016878 (787.7 Mb)  TX bytes:5745933 (5.4 Mb)
          Interrupt:19 Base address:0xb880 Memory:fe0fe000-fe0fe038

eth1      Link encap:Ethernet  HWaddr 00:81:8A:10:5C:34
          inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:19 Base address:0xef00 Memory:febfd000-febfd038

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:162 errors:0 dropped:0 overruns:0 frame:0
          TX packets:162 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11692 (11.4 Kb)  TX bytes:11692 (11.4 Kb)
```

This example shows that two ethernet interfaces (`eth0` and `eth1`) are present and running (as indicated by `UP` in the third line of each interface description.

If the second network does not appear, it may be that a network interface card must be installed in order to provide a second network, or it may be that the network is not yet initialized.

Using CXFS GUI Connectivity Diagnostics for Linux

In order to test node connectivity by using the GUI, the `root` user on the node running the CXFS diagnostics must be able to access a remote shell using the `rsh` command (as `root`) on all other nodes in the cluster. (This test is not required when using `cxfs_admin` because it verifies the connectivity of each node as it is added to the cluster.)

There are several ways of accomplishing this, depending on the existing settings in the pluggable authentication modules (PAMs) and other security configuration files.

The following method works with default settings. Do the following on all nodes in the cluster:

1. Install the `rsh-server` RPM.
2. Enable `rsh`.
3. Restart `xinted`.
4. Add `rsh` to the `/etc/securetty` file.
5. Add the hostname of the node from which you will be running the diagnostics into the `/root/.rhosts` file. Make sure that the mode of the `.rhosts` file is set to `600` (read and write access for the owner only).

After you have completed running the connectivity tests, you may wish to disable `rsh` on all cluster nodes.

For more information, see the Linux operating system documentation about PAM and the `hosts.equiv` man page.

Verifying the Private and Public Networks for Linux

For each private network on each Linux node in the pool, verify access with the `ping` command:

1. Enable multicast `ping` using one or more of the following methods (the permanent method will not take affect until after a reboot):

   - Immediate but temporary method:

     ```
     linux# echo "0" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
     ```

     For more information, see http://kerneltrap.org/node/16225

   - Immediate but temporary method:

     ```
     linux# sysctl net.ipv4.icmp_echo_ignore_broadcasts=0"
     ```

   - Permanent method upon reboot (survives across reboots):

     1. Remove the following line (if it exists) from the `/etc/sysctl.conf` file:

        ```
        net.ipv4.icmp_echo_ignore_broadcasts = 1
        ```

2. Execute a `ping` using the private network. Enter the following, where *nodeIPaddress* is the IP address of the node:

   ```
   # ping nodeIPaddress
   ```

   For example:

   ```
   linux# ping 10.0.0.1
   PING 10.0.0.1 (10.0.0.1) from 128.162.240.141 : 56(84) bytes of data.
   64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.310 ms
   64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.122 ms
   64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.127 ms
   ```

3. Execute a `ping` using the public network.

4. If the `ping` fails, repeat the following procedure on each node:

   a. Verify that the network interface was configured up using `ifconfig`. For example:

   ```
   linux# ifconfig eth1
   eth1      Link encap:Ethernet  HWaddr 00:81:8A:10:5C:34
             inet addr:10.0.0.10  Bcast:10.0.0.255  Mask:255.255.255.0
             UP BROADCAST MULTICAST  MTU:1500  Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:100
             RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
             Interrupt:19 Base address:0xef00 Memory:febfd000-febfd038
   ```

   In the third output line above, `UP` indicates that the interface was configured up.

   b. Verify that the cables are correctly seated.

5. Repeat this procedure on each node.

Client Software Installation for Linux

This section discusses the following:

Installing CXFS Software for Linux

Table 3-1 provides examples of the differences in package extensions among the various processor architectures supported by CXFS.

Note: The package extensions vary by architecture. Ensure that you install the appropriate package for your processor architecture.

Table 3-1 Processor Architecture and Package Extensions

Class	Processor Architecture	Package Architecture Extension
x86_64	AMD Opteron	`.x86_64.rpm`
	Intel Xeon EM64T	`.x86_64.rpm`
ia64	Intel Itanium 2	`.ia64.rpm`

Installing the CXFS client software for Linux requires approximately 50–200 MB of space, depending upon the packages installed at your site.

To install the required software on a Linux node, do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes, CXFS general release notes, and CXFS Linux release notes in the `/docs` directory on the ISSP DVD and any late-breaking caveats on the download page.
2. Verify that the node is running a supported Linux distribution and kernel, according to the CXFS for Linux release notes. See the Red Hat `/etc/redhat-release` or SLES `/etc/SuSE-release` files and enter the following:

   ```
   linux_cxfsclient# uname -r
   ```

3. *(Optional)* Verify that the node is running the supported level of software, according to the CXFS for Linux release notes. Also install any required patches. See the `releasenotes/README` file for more information.
4. If you had to install software in one of the above steps, reboot the system:
   - RHEL 7 or SLES 12:

     ```
     linux_cxfsclient# systemctl reboot
     ```

   - Earlier versions of RHEL and SLES:

     ```
     earlierlinux# /sbin/reboot
     ```

5. Transfer the client-only software (that was downloaded onto a CXFS server-capable administration node during its installation procedure) from the server to the client using `ftp`, `rcp`, or `scp`.

The location of the tarball on the server will be as follows:

/usr/cluster/client-dist/*CXFS_VERSION*/linux/*CLIENT_LINUX_VERSION*/*ARCHITECTURE*/cxfs-client.tar.gz

For example, for an SGI ia64 client running SLES 10 SP2, the location of the CXFS 6.1 tarball on the server would be:

/usr/cluster/client-dist/6.1.0.3/linux/sles10sp2/ia64/cxfs-client.tar.gz

---

Note: Specific packages listed here are examples and may not match the released product.

In this case, you could do the following:

```
server_admin# cd /usr/cluster/client-dist/6.1.0.3/linux/sles10sp2/ia64
server_admin# scp cxfs-client.tar.gz linux_cxfsclient:/tmp/cxfs/
```

6. Disassemble the downloaded tarball on the Linux client-only node. For example:

   ```
   linux_cxfsclient# cd /tmp/cxfs
   linux_cxfsclient# tar -zxvf tarball
   ```

   After you extract the information using `tar`, the RPMs will be in the following directory:

   ```
   /tmp/cxfs/sgi-install/SGI/RPMS
   ```

7. Install the CXFS software (line breaks shown for readability):

   - RHEL 5, RHEL 6, and RHEL 7:
     - Base installation (all systems):

       ```
       rhel# rpm -Uvh kmod*rpm cxfs_admin*rpm cxfs_client*rpm cxfs_util*rpm \
       sgidbg*rpm *commands*rpm
       ```

     - Add GRIOv2 support:

       ```
       rhel# rpm -Uvh grio2*rpm
       ```

     - Add PCP PMDAs support (supported platforms):

       ```
       rhel#  rpm -Uvh *pmda*rpm
       ```

- SLES 11 and SLES 12:
  - Base installation (all systems):

```
sles11# rpm -Uvh *kmp*rpm cxfs_admin*rpm cxfs_client*rpm cxfs_util*rpm \
sgidbg*rpm *commands*rpm
```

  - Add GRIOv2 support:

```
sles11# rpm -Uvh grio2*rpm
```

  - Add PCP PMDAs support (supported platforms):

```
sles11# rpm -Uvh *pmda*rpm
```

- SLES 10:
  - Base installation (all systems) for the `default` kernel:

```
sles10# rpm -Uvh *kmp-default*rpm cxfs_admin*rpm cxfs_client*rpm cxfs_util*rpm \
sgidbg*rpm *commands*rpm
```

  - Base installation (all systems) for the `smp` kernel:

```
sles10# rpm -Uvh *kmp-smp*rpm cxfs_admin*rpm cxfs_client*rpm cxfs_util*rpm \
sgidbg*rpm *commands*rpm
```

  - Add GRIOv2 support:

```
sles10# rpm -Uvh grio2*rpm
```

8. Edit the `/etc/cluster/config/cxfs_client.options` file as necessary. See the "Maintenance for Linux" on page 47 and the `cxfs_client`(8) man page.
9. Reboot the system:
   - RHEL 7 or SLES 12:

```
linux_cxfsclient# systemctl reboot
```

   - Earlier versions of RHEL and SLES:

```
earlierlinux# /sbin/reboot
```

10. *(RHEL 5 x86_64 systems only)* edit the `/etc/depmod.d/depmod.conf.dist` file and make the following change:

From:

```
search updates extra built-in weak-updates
```

To:

```
search updates extra weak-updates built-in
```

Verifying the Linux Installation

Use the `uname -r` command to ensure the kernel installed above is running.

To verify that the CXFS software has been installed properly, use the `rpm -qa` command to display all of the installed packages. You can filter the output by searching for particular package name.

Postinstallation Steps for Linux

This section discusses the following:

- "Configuring XVM Failover V2 on Linux" on page 44
- "Configuring I/O Fencing for Linux" on page 45

Configuring XVM Failover V2 on Linux

You can create the `/etc/failover2.conf` file by using the `mk_failover2`(8) command. For details, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage.*

If the RAID supports the asymmetric logical unit access (ALUA) feature, XVM will automatically read the RAID's settings for preferred controller for each LUN.

If ALUA is not supported, you should add a line with the keyword `preferred` to the `/etc/failover2.conf` file to configure the preferred controller. (Only non-ALUA RAID paths must be defined in the `/etc/failover2.conf` file.)

Following is an example of the `/etc/failover2.conf` file on a Linux system:

```
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200800a0b8184c8e:0x0000000000000000 affinity=0 preferred
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200900a0b8184c8d:0x0000000000000000 affinity=1
```

Configuring I/O Fencing for Linux

I/O fencing is required on Linux nodes in order to protect data integrity of the filesystems in the cluster. The `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Linux nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

Start/Stop `cxfs_client` for Linux

The `cxfs_client` service will be invoked automatically during normal system startup and shutdown procedures. This script starts and stops the `cxfs_client` daemon.

To start up `cxfs_client` manually, enter the following:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl start cxfs_client
  ```

- Earlier versions of RHEL and SLES:

  ```
  earlierlinux# service cxfs_client start
  ```

To stop `cxfs_client` manually, enter the following:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl stop cxfs_client
  ```

- Earlier versions of RHEL or SLES:

  ```
  linux# service cxfs_client stop
  ```

To stop and then start `cxfs_client` manually, enter the following:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl restart cxfs_client
  ```

- Earlier versions of RHEL or SLES:

  ```
  linux# service cxfs_client restart
  ```

To see the current status, use the `status` argument:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl status cxfs_client
  ```

- Other supported versions of RHEL or SLES:

  ```
  earlierlinux# service cxfs_client status
  ```

The following shows status for RHEL 7 or SLES 12:

```
rhel7_or_sles12# systemctl status cxfs_client
cxfs_client.service - CXFS client daemon
   Loaded: loaded (/usr/lib/systemd/system/cxfs_client.service; enabled)
   Active: active (exited) since Tue 2015-07-07 07:04:04 CDT; 2min 30s ago
     Docs: man:cxfs_client(1)
  Process: 42679 ExecStop=/etc/init.d/cxfs_client stop (code=exited, status=0/SUCCESS)
  Process: 42697 ExecStart=/etc/init.d/cxfs_client start (code=exited, status=0/SUCCESS)
  Main PID: 42697 (code=exited, status=0/SUCCESS)

Jul 07 07:04:04 node1.mycompany.com cxfs_client[42697]: Loading CXFS modules: [  OK  ]
Jul 07 07:04:04 node1.mycompany.com cxfs_client[42697]: Starting cxfs client: cxfs_client daemon started
Jul 07 07:04:04 node1.mycompany.com cxfs_client[42697]: [  OK  ]
Jul 07 07:04:04 node1.mycompany.com cxfs_client[42697]: cxfs_client daemon started
Jul 07 07:04:04 node1.mycompany.com cxfs_client[42697]: [  OK  ]
Jul 07 07:04:04 node1.mycompany.com systemd[1]: Started CXFS client daemon.
```

The following examples show status for earlier versions:

```
earlierlinux# service cxfs_client status
cxfs_client status [timestamp Apr 20 14:54:30 / generation 4364]

CXFS client:
    state: stable (5), cms: up, xvm: up, fs: up
Cluster:
    mycluster (23) - enabled
Local:
    node2 (7) - enabled
Nodes:
    node3      enabled  up    12
    node4      enabled  DOWN  10
    node5      enabled  up    11
    node6      enabled  up    7
```

```
    node7      enabled  up    4
    node8      enabled  up    9
    node9      enabled  up    5
    node10     enabled  up    8
    node11     enabled  up    2
    node12     enabled  up    0
    node13     enabled  up    6
    node14     enabled  up    3
    node15     enabled  up    1
Filesystems:
    concatfs   enabled  mounted          concatfs             /concatfs
    stripefs   enabled  mounted          stripefs             /stripefs
    tp9300_stripefs enabled  forced mounted   tp9300_stripefs      /tp9300_stripefs
cxfs_client is running.

earlierlinux# service cxfs_client status
cxfs_client is stopped
```

Maintenance for Linux

This section contains information about maintenance procedures for CXFS on Linux:

- "Modifying the CXFS Software for Linux" on page 47
- "Recognizing Storage Changes for Linux" on page 48

Modifying the CXFS Software for Linux

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/etc/cluster/config/cxfs_client.options` file. The available options are documented in the `cxfs_client` man page.

Caution: Some of the options are intended to be used internally by SGI only for testing purposes and do not represent supported configurations. Consult your SGI service representative before making any changes.

To see if `cxfs_client` is using the options in `cxfs_client.options`, enter the following:

```
linux# ps -ax | grep cxfs_client
 3612 ?        S      0:00 /usr/cluster/bin/cxfs_client -i cxfs3-5
 3841 pts/0    S      0:00 grep cxfs_client
```

To be sure that `cxfs_client` is configured to start up on boot, do the following:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl is-enabled cxfs_client
  enabled
  ```

- Other supported versions of RHEL or SLES:

  ```
  earlierlinux# chkconfig --list | grep cxfs_client
  cxfs_client              0:off  1:off  2:off  3:on   4:off  5:on   6:off
  ```

Recognizing Storage Changes for Linux

On Linux nodes, the `cxfs-enumerate-wwns` script enumerates the world wide names (WWNs) on the host that are known to CXFS. See "Mount Scripts" on page 28.

The following script is run by `cxfs_client` when it reprobes the storage controllers upon joining or rejoining membership:

```
/var/cluster/cxfs_client-scripts/cxfs-reprobe
```

For RHEL nodes, you can define a group of environment variables in the `/etc/cluster/config/cxfs_client.options` file in order for `cxfs-reprobe` to probe specific targets on the SCSI bus.

The script detects the presence of the SCSI and/or XSCSI layers on the system and defaults to probing whichever layers are detected. You can override this decision by setting `CXFS_PROBE_SCSI` and/or `CXFS_PROBE_XSCSI` to one of the following on the appropriate bus:

- `0` to disable the probe
- `1` to force the probe

When an XSCSI scan is performed, all buses are scanned by default. You can override this decision by specifying a space-separated list of buses in

CXFS_PROBE_XSCSI_BUSES. (If you include space, you must enclose the list within single quotation marks.) For example:

```
export CXFS_PROBE_XSCSI_BUSES='/dev/xscsi/pci0001:00:03.0-1/bus /dev/xscsi/pci0002:00:01.0-2/bus'
```

When a SCSI scan is performed, a fixed range of buses/channels/IDs and LUNs are scanned; these ranges may need to be changed to ensure that all devices are found. The ranges can also be reduced to increase scanning speed if a smaller space is sufficient.

The following summarizes the environment variables (separate multiple values by white space and enclose within single quotation marks):

CXFS_PROBE_SCSI=*0|1*

Stops (0) or forces (1) a SCSI probe. Default: 1 if SCSI

CXFS_PROBE_SCSI_BUSES=*BusList*

Scans the buses listed. Default: 0 1 2

CXFS_PROBE_SCSI_CHANNELS=*ChannelList*

Scans the channels listed. Default: 0

CXFS_PROBE_SCSI_IDS=*IDList*

Scans the IDS listed. Default: 0 1 2 3

CXFS_PROBE_SCSI_LUNS=*LunList*

Scans the LUNs listed. Default: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

CXFS_PROBE_XSCSI=*0|1*

Stops (1) or forces (1) an XSCSI probe. Default: 1 if XSCSI

CXFS_PROBE_XSCSI_BUSES=*BusList*

Scans the buses listed. Default: all XSCSI buses

For example, the following would only scan the first two SCSI buses:

```
export CXFS_PROBE_SCSI_BUSES='0 1'
```

The following would scan 16 LUNs on each bus, channel, and ID combination (all on one line):

```
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15'
```

Other options within the `/etc/cluster/config/cxfs_client.options` file begin with a `-` character. Following is an example `cxfs_client.options` file:

```
# Example cxfs_client.options file
#
-Dnormal -serror
export CXFS_PROBE_SCSI_BUSSES=1
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'
```

Note: The `-` character or the term `export` must start in the first position of each line in the `cxfs_client.options` file; otherwise, they are ignored by the `cxfs_client` service.

Using `cxfs-reprobe` with RHEL

When `cxfs_client` rescans disk buses, it executes the `/var/cluster/cxfs_client-scripts/cxfs-reprobe` script. This requires the use of parameters in RHEL due to limitations in the SCSI layer. You can export these parameters from the `/etc/cluster/config/cxfs_client.options` file.

The script detects the presence of the SCSI and/or XSCSI layers on the system and defaults to probing whichever layers are detected. You can override this decision by setting `CXFS_PROBE_SCSI` (for Linux SCSI) or `CXFS_PROBE_XSCSI` (for Linux XSCSI) to one of the following:

- `0` to disable the probe
- `1` to force the probe

When an XSCSI scan is performed, all buses are scanned by default. You can override this by specifying a space-separated list of buses in `CXFS_PROBE_XSCSI_BUSES`. (If you include space, you must enclose the list within single quotation marks.) For example:

```
export CXFS_PROBE_XSCSI_BUSES='/dev/xscsi/pci01.03.0-1/bus /dev/xscsi/pci02.01.0-2/bus'
```

When a SCSI scan is performed, a fixed range of buses/channels/IDs and LUNs are scanned; these ranges may need to be changed to ensure that all devices are found. The ranges can also be reduced to increase scanning speed if a smaller space is sufficient.

The following summarizes the environment variables (separate multiple values by white space and enclose within single quotation marks):

`CXFS_PROBE_SCSI=`*0|1*

Stops (`0`) or forces (`1`) a SCSI probe. Default: `1` if SCSI

`CXFS_PROBE_SCSI_BUSES=`*BusList*

Scans the buses listed. Default: `0 1 2`

`CXFS_PROBE_SCSI_CHANNELS=`*ChannelList*

Scans the channels listed. Default: `0`

`CXFS_PROBE_SCSI_IDS=`*IDList*

Scans the IDS listed. Default: `0 1 2 3`

`CXFS_PROBE_SCSI_LUNS=`*LunList*

Scans the LUNs listed. Default: `0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15`

`CXFS_PROBE_XSCSI=`*0|1*

Stops (`1`) or forces (`1`) an XSCSI probe. Default: `1` if XSCSI

`CXFS_PROBE_XSCSI_BUSES=`*BusList*

Scans the buses listed. Default: all XSCSI buses

For example, the following would only scan the first two SCSI buses:

```
export CXFS_PROBE_SCSI_BUSES='0 1'
```

The following would scan 16 LUNs on each bus, channel, and ID combination (all on one line):

```
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15'
```

Other options within the `/etc/cluster/config/cxfs_client.options` file begin with a `-` character. Following is an example `cxfs_client.options` file:

```
# Example cxfs_client.options file
#
-Dnormal -serror
export CXFS_PROBE_SCSI_BUSSES=1
export CXFS_PROBE_SCSI_LUNS='0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20'
```

Note: The `-` character or the term `export` must start in the first position of each line in the `cxfs_client.options` file; otherwise, they are ignored by the `cxfs_client` service.

GRIO on Linux

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Linux platform if GRIO is enabled on the server-capable administration node. However, GRIO is disabled by default on Linux client-only nodes. To enable GRIO on a Linux client-only node, you must install the GRIO software as documented in "Installing CXFS Software for Linux" on page 40 and do the following:

1. Change the following line in `/etc/cluster/config/cxfs_client.options` from:

   ```
   export GRIO2=off
   ```

   to:

   ```
   export GRIO2=on
   ```

2. Reboot the system.

A Linux node can mount a GRIO-managed filesystem and supports node-level static reservations. A Linux node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

Application bandwidth reservations must be explicitly released by the application before exit. If the application terminates unexpectedly or is killed, its bandwidth reservations are not automatically released and will cause a bandwidth leak. If this happens, the lost bandwidth could be recovered by rebooting the node.

For more information, see:

- "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7
- *Guaranteed-Rate I/O Version 2 for Linux Guide*

System Tunable Kernel Parameters on Linux

SGI recommends that you use the same settings for system tunable kernel parameters on all applicable nodes in the cluster.

Caution: Before changing any parameter, you should understand the ramifications of doing so on your system. You should only modify debugging parameters at the recommendation of SGI.

This section discusses the following:

- "Making Permanent Parameter Changes on Linux" on page 53
- "Making Temporary Parameter Changes on Linux" on page 54
- "Querying a Current Parameter Setting on Linux" on page 55
- "Parameter Details for Linux" on page 55

Making Permanent Parameter Changes on Linux

You can change a parameter permanently across reboots by adding it to the following file: `/etc/modprobe.d/sgi-cxfs-xvm.conf`

Note: You may have to create this file when adding the first parameter.

Use the following format:

`options` *module systune=setting*

where:

- *module* is one of the following module strings:

```
sgi-cell
sgi-cxfs
```

- *systune* is the parameter name, such as `mtcp_hb_watchdog`
- *setting* is the value you want to set for the parameter, such as `2`

Note: Do not use spaces around the = character.

For example, to permanently set the `mtcp_hb_watchdog` parameter (which is in the `sgi-cell` module) to `2`, add the following line to the configuration file:

```
options sgi-cell mtcp_hb_watchdog=2
```

The change will take effect upon reboot.

Making Temporary Parameter Changes on Linux

For a temporary change to a dynamic parameter, use the Linux `sysctl`(8) command:

```
linux# sysctl -w prefix.systune=value
```

where:

- *prefix* is one of the following:

  ```
  fs.cxfs
  kernel.cell
  ```

- *systune* is the parameter name, such as `mtcp_hb_watchdog`
- *setting* is the value you want to set for the parameter, such as `2`

Note: Do not use spaces around the = character.

For example, to set the `mtcp_hb_watchdog` parameter (which has the `kernel.cell` prefix) to `2`:

```
linux# sysctl -w kernel.cell.mtcp_hb_watchdog=2
kernel.cell.mtcp_hb_watchdog = 2
```

Querying a Current Parameter Setting on Linux

To query the current setting of a parameter on a Linux system, use the Linux`sysctl`(8) command:

```
linux# sysctl prefix.systune
```

where:

- *prefix* is one of the following:

  ```
  fs.cxfs
  kernel.cell
  ```

- *systune* is the parameter name, such as `mtcp_hb_watchdog`

For example, to query the current setting of the `mtcp_hb_watchdog` parameter (which has the `kernel.cell` prefix):

```
linux# sysctl kernel.cell.mtcp_hb_watchdog
kernel.cell.mtcp_hb_watchdog = 2
```

Parameter Details for Linux

For details about the available parameters, see the system tunable kernel parameter appendix in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Troubleshooting for Linux

This section discusses the following:

Also see:

Device Filesystem Enabled for Linux

The kernels provided for the Linux node have the Device File System (`devfs`) enabled. This can cause problems with locating system devices in some circumstances. See the `devfs` FAQ at the following location:

http://www.atnf.csiro.au/people/rgooch/linux/docs/devfs.html

The cxfs_client Daemon is Not Started on Linux

Confirm that the cxfs_client is not running. The following command would list the cxfs_client process if it were running:

```
linux# ps -ax | grep cxfs_client
```

Check the cxfs_client log file for errors.

Restart cxfs_client as described in "Start/Stop cxfs_client for Linux" on page 45 and watch the cxfs_client log file for errors.

To be sure that cxfs_client is configured to start up on boot, to the following:

- RHEL 7 or SLES 12:

  ```
  rhel7_or_sles12# systemctl is-enabled cxfs_client
  enabled
  ```

- Earlier Linux:

  ```
  earlierlinux# chkconfig --list | grep cxfs_client
  cxfs_client              0:off  1:off  2:off  3:on   4:off  5:on   6:off
  ```

Filesystems Do Not Mount on Linux

If cxfs_info reports that cms is up but XVM or the filesystem is in another state, then one or more mounts is still in the process of mounting or has failed to mount.

The CXFS node might not mount filesystems for the following reasons:

- The node may not be able to see all of the LUNs. This is usually caused by misconfiguration of the HBA or the SAN fabric:
  - Check that the ports on the Fibre Channel, SAS, or InfiniBand switch connected to the HBA are active. Physically look at the switch to confirm the light next to the port is green, or remotely check by using the switchShow command.
  - Check that the HBA configuration is correct.
  - Check that the HBA can see all the LUNs for the filesystems it is mounting.
  - Check that the operating system kernel can see all the LUN devices.
  - If the RAID device has more than one LUN mapped to different controllers, ensure the node has a SAN path to all relevant storage controllers.

- The `cxfs_client` daemon may not be running. See "The `cxfs_client` Daemon is Not Started on Linux" on page 57.
- The filesystem may have an unsupported mount option. Check the `cxfs_client.log` for mount option errors or any other errors that are reported when attempting to mount the filesystem.
- The cluster membership (`cms`), XVM, or the filesystems may not be up on the node. Execute the `cxfs_info` command to determine the current state of `cms`, XVM, and the filesystems. If the node is not up for each of these, then check the `/var/log/cxfs_client` log to see what actions have failed.

  Do the following:

  – If `cms` is not up, check the following:

    • Is the node is configured on the server-capable administration node with the correct hostname?

    • Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 205.

  – If XVM is not up, check that the HBA is active and can see the LUNs.

  – If the filesystem is not up, check that one or more filesystems are configured to be mounted on this node and check the `/var/log/cxfs_client` file for mount errors.

Unable to use the `dmi` Mount Option

By default, DMAPI is turned off on SLES 10 and SLES 11 systems. If you try to mount with the `dmi` mount option, you will see errors such as the following:

```
kernel: XFS: unknown mount option [dmi]."
```

See "Enable DMAPI for SLES 10 and SLES 11 Client-Only Nodes" on page 33.

Large Log Files on Linux

The `/var/log/cxfs_client` log file may become quite large over a period of time if the verbosity level is increased. See the `cxfs_client.options` man page and "Log Files on Linux" on page 28.

`xfs off` Output from `chkconfig`

The following output from `chkconfig --list` refers to the X Font Server, not the XFS filesystem, and has no association with CXFS:

```
xfs                     0:off  1:off  2:off  3:off  4:off  5:off  6:off
```

`crash` Dumps

To enable the collection of `crash` dumps on a Linux client-only node, consult your operating system documentation. The *CXFS 7 Administrator Guide for SGI InfiniteStorage* contains a procedure for enabling `crash` dump collection on a server-capable administration node.

Slow Performance on Linux Due to Token Optimizations

Note: You should modify `cell_tkm_feature_disable` only if directed to do so by SGI Support.

CXFS token prefetch and range tokens are designed as optimizations for applications using CXFS filesystems on a CXFS client. However, under some workloads, token prefetch may actually slow performance and range tokens may cause token hangs. If directed to do so by SGI Support, you can use the `cell_tkm_feature_disable` system tunable parameter to disable these features on Linux clients:

- To disable token prefetch:

  ```
  linux# sysctl -w kernel.cell.cell_tkm_feature_disable=4
  ```

- To disable range tokens:

  ```
  linux# sysctl -w kernel.cell.cell_tkm_feature_disable=64
  ```

- To disable both token prefetch and range tokens:

  ```
  linux# sysctl -w kernel.cell.cell_tkm_feature_disable=68
  ```

- To reenable both token prefetch and range tokens (returning to the default behavior):

  ```
  linux# sysctl -w kernel.cell.cell_tkm_feature_disable=0
  ```

For information about setting `cell_tkm_feature_disable` permanently, see "Making Permanent Parameter Changes on Linux" on page 53. For more details about system tunable parameters, see *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

SGI ia64 NMI System Reset Hangs

On SGI ia64 machines, an NMI system reset will not proceed until there is human intervention.

Note: NMI should only be used when directed to by SGI Service personnel. When used on SGI ia64 client-only nodes, the node will not restart automatically, but will stop in the `kdb` debugger, which requires human intervention to perform debugging and reset the node manually. (See SGI Bulletin TIB 200908 for information on debugging with `kdb`.)

Multiple Ethernet Interfaces on SGI ia64 Systems

In SGI ia64 systems with multiple Ethernet interfaces, the default behavior of the operating system is to dynamically assign interface names (such as `eth0`, `eth1`, and so on) at boot time. Therefore, the physical interface associated with the `eth0` device may change after a system reboot; if this occurs, it will cause a networking problem for CXFS. To avoid this problem, provide persistent device naming by using the `/etc/sysconfig/networking/eth0_persist` file to map specific Ethernet device names to specific MAC addresses. Adding lines of the format to the `eth0_persist` file:

`eth`*N MAC_ID*

For example:

```
eth0 08:00:69:13:dc:ec
eth1 08:00:69:13:72:e8
```

System Dump Analysis Tool (SLES 11 on ia64)

For system dump analysis on an SGI ia64 system running SLES 11, use the SLES `crash` tool. (For information about SLES x86_64 systems and RHEL systems, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.)

To enable the collection of crash dumps on a SLES 11 ia64 system, do the following:

1. Install the following RPMs, where *kernelrev* matches your installed kernel

   - `kernel-default-debuginfo-`*kernelrev*
   - `kdump-`*version*
   - `kexec-tools-`*version*

   For example, for the `2.6.27.19-5-default` kernel, you would require `kernel-default-debuginfo-2.6.27.19-5.1` RPM, which would install the following file:

   `/usr/lib/debug/boot/vmlinux-2.6.27.19-5-default.debug`

   When you install the `kdump-`*version* RPM, it will automatically add the following information onto the `kernel` lines in the `/etc/elilo.conf` file :

   ```
   crashkernel=512M-:256M
   ```

   Note: When you install the `kdump` RPM, `kdump` is automatically enabled. (Unlike previous SLES releases, when you had to manual do a `chkconfig kdump on`.)

2. Run the following command to make the changes to `/etc/elilo.conf` take effect:

   ```
   ia64# elilo
   ```

3. Reboot, which activates the kernel and reserves the required memory. You will see the following message on the console:

   ```
   Loading kdump
   Regenerating kdump initrd ... done
   System Boot Control: The system has been set up
   ```

4. Verify that the machine is set up correctly by requesting an NMI from the console:

   ```
   ia64_console# echo "c">/proc/sysrq-trigger
   ```

   Note: If there are several old dump files, the oldest one might be deleted by this process.

For example:

```
ia64_console# echo "c">/proc/sysrq-trigger
SysRq : Trigger a crashdump
Initializing cgroup subsys cpuset
Initializing cgroup subsys cpu
...
(pages of output)
```

The key piece of information to look for are lines such as the following at the end of the output:

```
Saving dump using makedumpfile
--------------------------------------------------------------------------------
Copying data                       : [100 %]

The dumpfile is saved to /root/var/crash/2009-10-29-21:03/vmcore.

makedumpfile Completed.
--------------------------------------------------------------------------------
Generating README             Finished.
Copying System.map            Finished.
Copying kernel                Finished.
```

Then the machine will reboot normally.

5. Go to the `/var/crash` directory and look for the dump directories that named according to the date and time. Each date directory will contain the files required for analysis. For example:

```
ia64# cd /var/crash
ia64# ls
2009-10-13-21:02/  2009-10-26-15:55/
ia64# ls -1 2009-10-26-15:55
README.txt
System.map-2.6.27.19-5-default
vmlinux-2.6.27.19-5-default.gz
vmcore
```

For more information, see the `crash`(8) man page.

No WWPNs Detected for Linux

If no WWPNs are detected, the following message will be logged to the `/var/log/cxfs_client` file:

```
cis_get_hbas no local HBAs found - falling back to /etc/fencing.conf
```

If no WWPNs are detected, you can manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `/etc/fencing.conf` file enumerates the WWPNs for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for each HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

You must update the `/etc/fencing.conf` file whenever the HBA configuration changes, including the replacement of an HBA.

For dual-ported HBAs, the file must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit. For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following (comment lines begin with #):

```
# WWPN of the HBA installed on this system
#
2000000173002c0b
```

To configure fencing, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Reporting Linux Problems

Before reporting a problem to SGI, you should run the `cxfsdump`(8) command on a server-capable administration node. See the information in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Note: You could run cxfsdump on each cluster node manually by executing the `/usr/cluster/bin/cxfsdump -local` command individually on each node, but this is less efficient.

Mac OS X Platform

CXFS supports a client-only node running the Mac OS X operating systems as defined in the CXFS Mac OS X release notes. This chapter contains the following sections:

CXFS on Mac OS X

This section contains the following information about CXFS on Mac OS X:

Requirements for Mac OS X

In addition to the items listed in "Requirements" on page 6, using a Mac OS X node to support CXFS requires the following, as detailed in the Mac OS X release notes:

- A supported Mac OS X operating system
- A supported Apple Computer hardware platform
- A supported Apple host bus adapter (HBA)

CXFS Commands on Mac OS X

The following commands are shipped as part of the CXFS Mac OS X package:

```
/usr/cluster/bin/autopsy
/usr/cluster/bin/cxfs
/usr/cluster/bin/cxfs_admin
/usr/cluster/bin/cxfs_client
/usr/cluster/bin/cxfs_info
/usr/cluster/bin/cxfscp
/usr/cluster/bin/cxfsdump
/usr/cluster/bin/fabric_dump
/usr/cluster/bin/frametest
/usr/cluster/bin/install-cxfs
/usr/cluster/bin/uninstall-cxfs
/usr/cluster/bin/xattr_convert
/usr/sbin/grioadmin
/usr/sbin/griomon
/usr/sbin/grioqos
/usr/cluster/bin/xvm
```

For more information on these commands, see the man pages and the *CXFS 7 Administrator Guide for SGI InfiniteStorage.*

Note the following:

- On Lion, Mountain Lion, Mavericks, and Yosemite, CXFS calls the `/usr/cluster/libexec/xvmfod` daemon at startup. XVM uses `xvmfod` to enable/disable paths to a RAID controller's WWNN by executing the `/usr/cluster/libexec/xvm_manage_paths` binary, which in turn calls `mpioutil` (1) to do the real work to manage the physical Fibre Channel paths to the LUN. `xvmfod` should be running on the Mac OS X node at all times while CXFS and XVM are active, if it is not, see "XVM Failover Problems on Lion, Mountain Lion, Mavericks, and Yosemite Nodes" on page 98.

  Note: The `xvmfod` command and `xvm_manage_paths` binary are intended to be used by XVM only; do not execute them manually.

- The installation package uses `install-cxfs` to install or update all of the CXFS files. You can use the `uninstall-cxfs` command to uninstall all CXFS files; uninstall is not an installation package option.
- The `cxfs_client` and `xvm` commands are needed to include a client-only node in a CXFS cluster.
- The `cxfs` command is run by the operating system to start and stop CXFS on the Mac OS X node.
- The `cxfs_info` command reports the current status of this node in the CXFS cluster.
- To make administrative changes via `cxfs_admin` from a client-only node, you must first use the `cxfs_admin access` command on a server-capable administration node to grant `admin` permission to the client-only node. For more information, see the section about setting `cxfs_admin` access permissions in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.
- For additional information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and "GRIO on Mac OS X" on page 91.

- If a Mac OS X node panics, the OS will write details of the panic to a log file:

  ```
  /Library/Logs/DiagnosticReports/*.panic
  /var/db/PanicReporter/current.panic (symbolic link to latest *.panic)
  ```

  Running `autopsy` parses this file and adds symbolic backtraces where possible to make it easier to determine the cause of the panic. The `autopsy` script is automatically run as part of the `cxfsdump` script, so the recommended steps for gathering data from a problematic node are still the same. Run `autopsy` with the `-man` option to display the man page.

  To display details of all visible devices on the Fibre Channel fabric, run the `fabric_dump` script. The output is useful for diagnosing issues related to mount problems due to missing LUNs. Run `fabric_dump` with the `-man` option to display the man page.

Log Files on Mac OS X

The `cxfs_client` command creates a `/var/log/cxfs_client` log file. To rotate this log file, use the `-z` option in the `/usr/cluster/bin/cxfs_client.options` file; see the `cxfs_client` man page for details.

The CXFS installation process (`install-cxfs` and `uninstall-cxfs`) appends to `/var/log/cxfs_inst.log`.

For information about the log files created on server-capable administration nodes, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage.*

Also see the following Mac OS X files:

- Snow Leopard or Lion:

  ```
  /var/log/kernel.log
  /var/log/system.log
  ```

- Mountain Lion, Mavericks, and Yosemite:

  ```
  /var/log/system.log
  ```

Limitations and Considerations for Mac OS X

CXFS for Mac OS X has the following limitations and considerations:

- XVM volume names are limited to 31 characters and subvolumes are limited to 26 characters. For more information about XVM, see *XVM Volume Manager Administrator Guide*.
- CXFS does not support the Spotlight indexing facility or the Time Machine backup facility, because these activities are applicable only to a local filesystem.

See also Appendix B, "Filesystem and Logical Unit Specifications" on page 227.

Configuring Hostnames on Mac OS X

Normally, you specify the hostname by using the following menu selection:

System Preferences
> Sharing
> Computer Name

Although the `HOSTNAME=-AUTOMATIC-` entry does not exist in the `/etc/hostconfig` file, you can specify a hostname by using the `HOSTNAME` parameter in this file. The hostname specified for the machine will have the following domain by default:

```
.local
```

For example, if the hostname was specified as `cxfsmac1`, then you would see the following when requesting the hostname:

```
macosx# /bin/hostname
cxfsmac1.local
```

The full hostname including `.local` is the hostname that the CXFS software will use to determine its identity in the cluster, not `cxfsmac1`.

Therefore, you must configure the node as `cxfsmac1.local` or specify the fully qualified hostname in `/etc/hostconfig`. For example:

```
HOSTNAME=cxfsmac1.sgi.com
```

Specifying the hostname in this way may impact some applications, most notably Bonjour, and should be researched and tested carefully. There are also known issues

with the hostname being reported as `localhost` on some reboots after making such a change.

SGI recommends that you specify other hosts in the cluster by editing `/etc/hosts`.

Mapping User and Group Identifiers for Mac OS X

To ensure that the correct access controls are applied to users on Mac OS X nodes when accessing CXFS filesystems, you must ensure that the user IDs (UIDs) and group IDs (GIDs) are the same on the Mac OS X node as on all other nodes in the cluster, particularly any server-capable administration nodes.

Note: A user does not have to have user accounts on all nodes in the cluster. However, all access control checks are performed by server-capable administration nodes, so any server-capable administration nodes must be configured with the superset of all users in the cluster.

Users can quickly check that their UID and GID settings are correct by using the `id` command on both the Mac OS X node and the server-capable administration node. For example:

```
macosx% id
uid=1113(fred) gid=999(users) groups=999(users), 20(staff)

admin% id
uid=1113(fred) gid=999(users) groups=999(users), 20(staff)
```

If the UID and/or GID do not match, or if the user is not a member of the same groups, then the user may unexpectedly fail to access some files.

The **Accounts Preference Pane** hides a set of advanced options that you can use to customize user account settings. Do the following:

1. Control-click a name in the **Accounts Preference Pane**.
2. Choose **Advanced** from the pop-up menu.
3. Select the item you want to change.

Access Control Lists and Mac OS X

All CXFS files have POSIX mode bits (read, write, and execute) and optionally an access control list (ACL). For more information, see the `chmod` and `chacl` man pages on a server-capable administration node.

CXFS on Mac OS X supports both the enforcement of POSIX ACLs and the editing of POSIX ACLs from the Mac OS X node.

This section discusses the following:

Note: In the following examples, line breaks are shown here for readability.

Displaying ACLs

To display ACLs on a Mac OS X node, use the `ls -l` command. For example, the `+` character after the file permissions indicates that there are ACLs for `newfile`:

```
macosx# ls -l newfile
 -rw-r--r--+ 1 userA ptg 4 Jan 18 09:49 newfile
```

To list the ACLs in detail, use the `-le` options (line breaks shown here for readability):

```
macosx# ls -le newfile
-rw--wxr--+ 1 userA  ptg  4 Jan 18 09:49 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny read,readattr,readextattr,readsecurity
 3: group:ptg allow read,execute,readattr,readextattr,readsecurity
 4: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chown
 5: group:everyone allow read,readattr,readextattr,readsecurity
 6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,writesecurity,chown
```

Comparing POSIX ACLs with Mac OS X ACLs

POSIX ACLs (implemented by CXFS) are very different from those available on Mac OS X. Therefore a translation occurs, which places some limitations on what can be achieved with Mac OS X ACLs. As shown in Table 4-1, POSIX supports only three types of access permissions; in contrast, Mac OS X supports many variations. This means that some granularity is lost when converting between the two systems.

Table 4-1 Mac OS X Permissions Compared with POSIX Access Permissions

POSIX	Mac OS X
Read	Read data, read attributes, read extended attributes, read security
Write	Write data, append data, delete, delete child, write attributes, write extended attributes, write security, add file, add subdirectory, take ownership, linktarget, check immutable
Execute	Execute

POSIX ACLs and the file permissions have a particular relationship that must be translated to work with Mac OS X ACLs. For example, the minimum ACL for a file is user, group, and other, as follows:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--]
```

The ACL (user, group, and other) exactly matches the file permissions. Further, any changes to the file permissions will be reflected in the ACL, and vice versa. For example:

```
server-admin# chmod 167 newfile
admin# chacl -l newfile
newfile [u::--x,g::rw-,o::rwx]
```

This is slightly complicated by the mask ACL, which if it exists takes the file's group permissions instead. For example:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--,m::rwx]
```

With POSIX, it is not possible to have fewer than three ACL entries, which ensures the rules always match with the file permissions. On Mac OS X, ACLs and file permissions are treated differently. ACLs are processed first; if there is no matching rule, the file permissions are used. Further, each entry can either be an `allow` entry or a `deny` entry. Given these differences, some restrictions are enforced to allow translation between these systems. For example, the simplest possible Linux ACL:

```
server-admin# chacl -l newfile
newfile [u::rw-,g::r-x,o::r--]
```

And the comparative Mac OS X ACL:

```
macosx# ls -le newfile
-rw-r-xr--+ 1 userA  ptg  4 Jan 18 09:49 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:ptg allow read,execute,readattr,readextattr,readsecurity
 3: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chown
 4: group:everyone allow read,readattr,readextattr,readsecurity
 5: group:everyone deny write,execute,delete,append,writeattr,writeextattr,writesecurity,chown
```

Each POSIX rule is translated into two Mac OS X rules. For example, the following user rules are equivalent:

- Linux:

  ```
  u::rw-
  ```

- Mac OS X:

  ```
  0: user:userA allow read,write,delete,append,readattr,writeattr,
            readextattr,writeextattr,readsecurity,writesecurity,chown
  1: user:userA deny execute
  ```

However, because the mask rule limits the access that can be assigned to anyone except the owner, the mask is represented by a single `deny` rule. For example, the following are equivalent:

- Linux:

  ```
  linux# chacl -l newfile
  newfile [u::rw-,g::r-x,o::r--,m::-wx]
  ```

- Mac OS X:

```
macosx# ls -le newfile
-rw--wxr--+ 1 userA  ptg  4 Jan 18 09:49 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny read,readattr,readextattr,readsecurity
 3: group:ptg allow read,execute,readattr,readextattr,readsecurity
 4: group:ptg deny write,delete,append,writeattr,writeextattr,writesecurity,chown
 5: group:everyone allow read,readattr,readextattr,readsecurity
 6: group:everyone deny write,execute,delete,append,writeattr,writeextattr,
   writesecurity,chown
```

The mask rule (`m::-wx`) is inverted into a simple deny rule (`group:everyone deny read,readattr,readextattr,readsecurity`). If a mask rule exists, it is always rule number 2 because it applies to everyone except for the file owner.

Editing POSIX ACLs on Mac OS X

To add, remove, or edit a POSIX ACL on a file or directory, use the `chmod` command, which allows you to change only a single rule at a time.

However, it is not valid in POSIX to have a single entry in an ACL. Therefore the basic rules are created based on the file permissions. For example:

```
macosx# ls -le newfile
-rw-rw-rw-   1 userA  ptg  0 Jan 18 15:40 newfile
macosx# chmod +a "cxfs allow read,execute" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA  ptg  0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny execute
 3: user:cxfs allow read,execute,readattr,readextattr,readsecurity
 4: user:cxfs deny write,delete,append,writeattr,writeextattr,writesecurity,chown
 5: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 6: group:ptg deny execute
 7: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
```

```
 8: group:everyone deny execute
```

You should only ever add, modify, or remove the `allow` rules. The corresponding `deny` rule will be created, modified, or removed as necessary. The `mask` rule is the only `deny` rule that you should specify directly.

For example, to remove a rule by using `chmod`:

```
macosx# chmod -a# 3 newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA ptg 0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny execute
 3: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 4: group:ptg deny execute
 5: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 6: group:everyone deny execute
```

If you remove rules leaving only the user, group, and other rules, ACLs will be removed completely. For example:

```
macosx# chmod -a# 2 newfile
macosx# ls -le newfile
-rw-rw-rw- 1 userA ptg 0 Jan 18 15:40 newfile
```

Adding rules to an existing ACL is complicated slightly because the ordering required by CXFS is different from the order used on Mac OS X. You may see the following error:

```
macosx# chmod +a "cxfs allow execute" newfile
chmod: The specified file newfile does not have an ACL in canonical order, please
specify a position with +a# : Invalid argument
```

However, because an order will be enforced regardless of where the rule is placed, insert at any position and the rules will be sorted appropriately. For example:

```
macosx# chmod +a# 6 "sshd allow execute" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA  ptg  0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
```

```
    readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny execute
 3: user:cxfs allow execute
 4: user:cxfs deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 5: user:sshd allow execute
 6: user:sshd deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 7: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 8: group:ptg deny execute
 9: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 10: group:everyone deny execute
```

You can also edit an existing rule by using `chmod`. Assuming the above file and permissions, you could allow the user to read files with the following command:

```
macosx# chmod =a# 3 "cxfs allow execute,read" newfile
macosx# ls -le newfile
-rw-rw-rw-+ 1 userA  ptg  0 Jan 18 15:40 newfile
 0: user:userA allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 1: user:userA deny execute
 2: group:everyone deny execute
 3: user:cxfs allow read,execute,readattr,readextattr,readsecurity
 4: user:cxfs deny write,delete,append,writeattr,writeextattr,writesecurity,chown
 5: user:sshd allow execute
 6: user:sshd deny read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 7: group:ptg allow read,write,delete,append,readattr,writeattr,readextattr,writeextattr,
    readsecurity,writesecurity,chown
 8: group:ptg deny execute
 9: group:everyone allow read,write,delete,append,readattr,writeattr,readextattr,
    writeextattr,readsecurity,writesecurity,chown
 10: group:everyone deny execute
```

Adding a second rule for the same user or group is not permitted with POSIX ACLs. If you attempt to do this, the permissions will be merged. It is important to get the rule number correct when editing a rule.

Default or Inherited ACLs on Mac OS X

It is possible to define default ACLs to a directory, so that all new files or directories created below are assigned a set of ACLs automatically. The semantics are handled differently between Linux and Mac OS X, so the functionality is limited to mimic what is available in POSIX. In POSIX, the default ACL is applied at creation time only; if the default rule subsequently changes, it is not applied to a directory's children. The equivalent behavior on Mac OS X is achieved by the `only_inherit` and `limit_inherit` flags.

For example, a default ACL might look like this on Linux:

```
admin# chacl -l test
test [u::rwx,g::r--,o::---/u::rw-,g::rw-,o::r--,u:501:r--,m::rwx]
```

On Mac OS X, a default ACL might look like the following:

```
macosx# ls -lde test
drwxr-----+ 2 userA  ptg  78 Jan 18 15:39 test
 0: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
    readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny
 2: group:ptg allow list,readattr,readextattr,readsecurity
 3: group:ptg deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
    writeextattr,writesecurity,chown
 4: group:everyone allow
 5: group:everyone deny list,add_file,search,delete,add_subdirectory,delete_child,
    readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown
 6: user:userA allow list,add_file,delete,add_subdirectory,delete_child,readattr,
    writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,file_inherit,
    directory_inherit,only_inherit
 7: user:userA deny search,file_inherit,directory_inherit,only_inherit
 8: group:everyone deny file_inherit,directory_inherit,only_inherit
 9: user:cxfs allow list,readattr,readextattr,readsecurity,file_inherit,
    directory_inherit,only_inherit
 10: user:cxfs deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
 11: group:ptg allow list,add_file,delete,add_subdirectory,delete_child,readattr,
     writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,file_inherit,
     directory_inherit,only_inherit
 12: group:ptg deny search,file_inherit,directory_inherit,only_inherit
 13: group:everyone allow list,readattr,readextattr,readsecurity,file_inherit,
     directory_inherit,only_inherit
```

```
 14: group:everyone deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
```

The default rules are flagged with the inheritance flags (`file_inherit,directory_inherit,only_inherit`). Editing these rules is similar to editing an access rule, except the inherit flag is included. For example:

```
macosx# mkdir newdir
macosx# chmod +a "cxfs allow read,only_inherit" newdir
macosx# ls -led newdir
drwxr-xr-x+ 2 userA  ptg  6 Jan 20 11:20 newdir
 0: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
    readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown
 1: user:userA deny
 2: group:ptg allow list,search,readattr,readextattr,readsecurity
 3: group:ptg deny add_file,delete,add_subdirectory,delete_child,writeattr,
    writeextattr,writesecurity,chown
 4: group:everyone allow list,search,readattr,readextattr,readsecurity
 5: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
    writeextattr,writesecurity,chown
 6: user:userA allow list,add_file,search,delete,add_subdirectory,delete_child,
    readattr,writeattr,readextattr,writeextattr,readsecurity,writesecurity,chown,
    file_inherit,directory_inherit,only_inherit
 7: user:userA deny file_inherit,directory_inherit,only_inherit
 8: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
    writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
 9: user:cxfs allow list,readattr,readextattr,readsecurity,file_inherit,
    directory_inherit,only_inherit
 10: user:cxfs deny add_file,search,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
 11: group:ptg allow list,search,readattr,readextattr,readsecurity,file_inherit,
     directory_inherit,only_inherit
 12: group:ptg deny add_file,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
 13: group:everyone allow list,search,readattr,readextattr,readsecurity,
     file_inherit,directory_inherit,only_inherit
 14: group:everyone deny add_file,delete,add_subdirectory,delete_child,writeattr,
     writeextattr,writesecurity,chown,file_inherit,directory_inherit,only_inherit
```

The base ACL is created if its not specified and removing the default ACL is a matter of removing rules until only the base rules are present, at which point the ACL will be removed.

HBA Installation for Mac OS X

CXFS for Mac OS X supports Apple Computer, Inc. host bus adapters (HBAs) and ATTO HBAs.

Note: The procedures in this section may be performed by you or by a qualified service representative. You must be logged in as `root` to perform the steps listed in this section.

This section discusses the following:

Installing the HBA for Mac OS X

Do the following:

1. Install the HBA into a spare PCI, PCI-X, or PCI Express slot in the Mac OS X node, according to the manufacturer's instructions. Do not connect the HBA to the Fibre Channel switch at this time.

   Note: Apple HBAs are normally shipped with copper SFPs and copper cables, so additional optic SFPs and optic cables may be required.

   ---

2. Reboot the node.

Installing the Fibre Channel Utility for Mac OS X

Do the following:

1. Install the configuration utility from the CD distributed with the HBA to your **Application** directory.
2. Run the Fibre Channel Utility after it is copied to the node. The tool will list the HBA on the left-hand side of the window. Select the card display the status of the

ports via a pull-down menu. Initially, each port will report that it is up (even though it is not connected to the switch), and the speed and port topology will configure automatically.

3. Connect one of the HBA ports to the switch via a Fibre Channel cable. After a few seconds, close and relaunch the Fibre Channel Utility. Select the card and then the connected port from the drop-down list to display the speed of the link.

   Repeat these steps for the second HBA port if required.

4. *(Optional)* For the Apple HBA, use Apple's `/sbin/fibreconfig` tool to modify port speed and topology if necessary. See the man page for details.

   The CXFS `fabric_dump` tool can also be of use in verifying Fibre Channel fabric configuration. See "CXFS Commands on Mac OS X" on page 66.

Configuring Two or More HBA Ports

The Mac OS X node does its own path management, but the operation varies by OS version:

- On Snow Leopard, the Mac OS X manages paths that go to the **same RAID controller** and only presents one `/dev` device to userspace per RAID controller. If multiple paths exist to a RAID controller, you will only see one `/dev` device.
- On Lion, Mountain Lion, Mavericks, and Yosemite, the Mac OS X node manages paths that go to the **same logical unit (LUN)** and only presents one `/dev` device to userspace per LUN. If multiple paths exist to a LUN (even across multiple RAID controllers), you will only see one `/dev` device.

Therefore, the Fibre Channel Utility does not support masking LUNs on specific ports. However, if the first port can see all of the LUNs, the default is that all I/O will go through a single port. To avoid this, configure the switch so that each port can see a different set of LUNs. You can achieve this by zoning the switch or by using multiple switches, with different controllers and HBA ports to each switch.

Note: To manage paths, you should use XVM. Manually enabling/disabling paths outside of XVM will break XVM failover and will produce unexpected results, and is therefore not supported by SGI.

Using `point-to-point` Fabric Setting for Apple HBAs

SGI recommends that you use the manual `point-to-point` fabric setting rather than rely on automatic detection, which can prove unreliable after a reboot.

Preinstallation Steps for Mac OS X

This section provides an overview of the steps that you or a qualified Apple service representative will perform on your Mac OS X nodes prior to installing the CXFS software. It contains the following sections:

- "Adding a Private Network for Mac OS X Nodes" on page 81
- "Verifying the Private and Public Networks for Mac OS X" on page 82
- "Disabling Power Saving Modes for Mac OS X" on page 83

Adding a Private Network for Mac OS X Nodes

The following procedure provides an overview of the steps required to add a private network to the Mac OS X system. A private network is required for use with CXFS. See "Use a Private Network" on page 13.

You may skip some steps, depending upon the starting conditions at your site. For details about any of these steps, see the Mac OS X system documentation.

1. Install Mac OS X and configure the machine's hostname (see "Configuring Hostnames on Mac OS X" on page 69) and IP address on its public network interface.
2. Add the IP addresses and hostnames of other machines in the cluster to the `/etc/hosts` file. You should be consistent about specifying the hostname or the fully qualified domain name for each host. A common convention is to name the CXFS private network address for each host as *hostname*`-priv`.
3. Install a second network interface card if necessary as per the manufacturer's instructions.

4. Configure the second network interface by using the following menu selection:

   System Preferences
   > Network
   > *(select the device for the second network and specify its information)*

   Select the second network interface (most likely `PCI Ethernet Slot 1`), and specify the IP address, subnet mask, and router. The private network interface should not require a DNS server because the private network address of other cluster nodes should be explicitly listed in the `/etc/hosts` file. Relying on a DNS server for private network addresses introduces another point of failure into the cluster and must be avoided.

5. Confirm the configuration using `ifconfig` to list the network interfaces that are up:

   ```
   macosx# ifconfig -u
   ```

   In general, this should include `en0` (the onboard Ethernet) and `en1` (the additional PCI interface), but the names of these interfaces may vary.

   For more information, see the `ifconfig` man page.

Verifying the Private and Public Networks for Mac OS X

Verify each interface by using the `ping` command to connect to the public and private network addresses of the other nodes that are in the CXFS pool.

For example:

```
macosx# grep cxfsmac2 /etc/hosts
134.14.55.115 cxfsmac2
macosx# ping -c 3 134.14.55.115
PING 134.14.55.115 (134.14.55.115): 56 data bytes
64 bytes from 134.14.55.115: icmp_seq=0 ttl=64 time=0.247 ms
64 bytes from 134.14.55.115: icmp_seq=1 ttl=64 time=0.205 ms
64 bytes from 134.14.55.115: icmp_seq=2 ttl=64 time=0.197 ms

--- 134.14.55.115 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 0.197/0.216/0.247 ms
```

Disabling Power Saving Modes for Mac OS X

Note the following:

- CXFS does not support the energy-saving mode on Mac OS X. If this mode is enabled, the Mac OS X node will lose CXFS membership and unmount the CXFS filesystem whenever it is activated.

  Select the following to disable the energy-saving mode:

  System Preferences
  > Energy Saver
  > Put the computer to sleep when it is inactive for
  > Never

- Clients connected to a DDN RAID should have disk sleep disabled. Uncheck the following selection:

  System Preferences
  > Energy Saver
  > Put the hard disk(s) to sleep when possible

- Never put CXFS clients to sleep. Select the following:

  System Preferences
  > Energy Saver
  > Put the computer to sleep when it is inactive
  > NEVER

Client Software Installation for Mac OS X

Installing the CXFS client software for Mac OS X requires approximately 30 MB of space.

To install the required software on a Mac OS X node, SGI personnel will do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes and CXFS release notes in the `/docs` directory on the ISSP DVD and late-breaking caveats on the download page.
2. Verify that the node is running a supported Mac OS X operating system according to the Mac OS X installation guide. Use the following command to display the currently installed system:

   ```
   macosx# uname -r
   ```

   This command should return a Darwin kernel value of:

   - `10.6.0` or later for Snow Leopard
   - `10.7.0` or later for Lion
   - `10.8.0` or later for Mountain Lion
   - `10.9.0` or later for Mavericks
   - `10.10.0` or later for Yosemite

3. As `root` or a user with administrative privileges, transfer the client software that was downloaded onto a server-capable node during its installation procedure using `ftp`, `rcp`, or `scp`. The location of the disk image on the server will be as follows:

`/usr/cluster/client-dist/`*CXFS_VERSION*`/macosx/`*MAC_VERSION*`/noarch/cxfs.dmg`

> **Note:** You must transfer the disk image to the `root` home directory (`/`) or your own home directory in order to make it visible with the **Finder** tool.

4. Double-click the transferred **cxfs.dmg** file to mount the disk image.
5. Double click **cxfs.pkg** to begin the installation.

6. Click **continue** when you see the following message:

   ```
   message : This package contains a program that determines
   if the software can be installed. Are you sure you want to continue
   ```

7. Click **continue** when you see the following message:

   ```
   The installer will guide you through the steps necessary to
    install CXFS for Mac OS X. To get started, click Continue
   ```

   This will launch the installation application, which will do the following:

   a. Display the CXFS Mac OS X release note. Read the release note and click **continue**.

   b. Display the license agreement. Read the agreement and click **agree** if you accept the terms.

   c. Perform a standard installation of the software on the root drive volume.

Caution: Do not choose **Change install location**.

8. Choose **Continue Installation** at the following message:

   ```
   Installation of this software requires you to restart your
   computer when the installation is done. Are you sure you want to
   install the software now?
   ```

9. After the install succeeds, click the highlighted **Restart** button to reboot your machine.

Postinstallation Steps for Mac OS X

This section discusses the following:

Configuring XVM Failover V2 on Mac OS X

This section discusses the following:

Failover for Mac OS X Lion and Later

For Mac OS X Lion and later (10.7.4 and later), if the RAID supports the asymmetric logical unit access (ALUA) feature, Mac OS X will automatically read the RAID's settings for preferred controller for each LUN.

If ALUA is not supported, you should add a line with the keyword `preferred` to the `/etc/failover2.conf` file to configure the preferred controller. (Only non-ALUA RAID paths must be defined in the `/etc/failover2.conf` file.)

On Lion and later, the kernel presents only one `/dev` device per LUN (as opposed to one per RAID controller), no matter how many physical paths are connected, even across multiple RAID controllers.

Nodes running Lion and later with non-ALUA RAID use the following format:

```
# node=1T42054894-200400a0b80cd5fe-000 LUN=0
200400a0b80cd5fe-000 preferred
# node=1T42054900-200500a0b80cd5fe-000 LUN=0
200500a0b80cd5fe-000 affinity=2

# node=1T42054894-200400a0b80cd5fe-001 LUN=1
200400a0b80cd5fe-001 affinity=2
# node=1T42054900-200500a0b80cd5fe-001 LUN=1
200500a0b80cd5fe-001 affinity=1 preferred
```

Following is an example of a set of XVM failover commands:

```
mymac# xvm foswitch -affinity 2 phys/lun0
[200400a0b81607f2-000=affinity1, 200500a0b81607f2-000=affinity2]
mymac# xvm foswitch -affinity 2 phys/lun2
[200400a0b81607f2-002=affinity1, 200500a0b81607f2-002=affinity2]
mymac# xvm foswitch -preferred phys/lun0
[200400a0b81607f2-000=affinity1, preferred]
```

Following are the corresponding messages that would appear in `/var/log/kernel.log`:

```
Apr 12 23:39:23 mymac kernel[0]: 11 0x83447f0 CXFS
<WARNING>xvm_fo_changepath: Enable wwnn[200500a0b81607f2] Disable wwnn[200400a0b81607f2] dev<17:1>
Apr 12 23:39:54 mymac kernel[0]: 26 0x826c000 CXFS
<WARNING>xvm_fo_changepath: Enable wwnn[200500a0b81607f2] Disable wwnn[200400a0b81607f2] dev<17:2>
Apr 12 23:40:02 mymac kernel[0]: 35 0x7b5b000 CXFS
<WARNING>xvm_fo_changepath: Enable wwnn[200400a0b81607f2] Disable wwnn[200500a0b81607f2] dev<17:1>
```

Failover for Mac OS X Snow Leopard

Following is an example of the `/etc/failover2.conf` file format for Mac OS X Snow Leopard:

```
/dev/rdisk-xvm-200400a0b80cd5fe-000 affinity=1 preferred
/dev/rdisk-xvm-200500a0b80cd5fe-000 affinity=2

/dev/rdisk-xvm-200400a0b80cd5fe-001 affinity=2
/dev/rdisk-xvm-200500a0b80cd5fe-001 affinity=1 preferred
```

The device is the node's WWNN plus the LUN number.

Note: On Snow Leopard, the kernel presents one `/dev` device per RAID controller, even if multiple paths exist. The node does its own path management for paths that go to the same RAID controller. See "Configuring Two or More HBA Ports" on page 80.

You can use the `mk_failover2`(8) command to create the `failover2.conf` file for CXFS Mac OS X nodes. For details, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Example `mk_failover2`(8) for Mac OS X

Following is an example of the `/etc/failover2.conf` file on a Linux system:

```
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200800a0b8184c8e:0x0000000000000000 affinity=0 preferred
/dev/disk/by-path/pci-0000:06:02.1-fc-0x200900a0b8184c8d:0x0000000000000000 affinity=1
```

Configuring I/O Fencing for Mac OS X

I/O fencing is required on Mac OS X nodes in order to protect data integrity of the filesystems in the cluster. The `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Mac OS X nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

Start/Stop `cxfs_client` for Mac OS X

The `/usr/cluster/bin/cxfs start` script will be invoked automatically during normal system startup and shutdown procedures. This script starts and stops the `cxfs_client` daemon.

To start `cxfs_client` manually, enter the following:

```
macosx# sudo /usr/cluster/bin/cxfs start
```

To stop `cxfs_client` manually, enter the following:

```
macosx# sudo /usr/cluster/bin/cxfs stop
```

To stop and start `cxfs_client` manually, enter the following:

```
macosx# sudo /usr/cluster/bin/cxfs restart
```

To prevent the automatic startup of `cxfs_client` on boot, edit the file `/Library/LaunchDaemons/com.sgi.cxfs.plist` to change the following setting:

- From:

  ```
  <key>RunAtLoad</key>
  <true/>
  ```

- To:

  ```
  <key>RunAtLoad</key>
  <false/>
  ```

Maintenance for Mac OS X

This section contains the following:

Updating the CXFS Software for Mac OS X

Do the following:

1. Ensure that no applications on the node are accessing files on a CXFS filesystem
2. Run the new CXFS software package, which will update all CXFS software.
3. Reboot.

Modifying the CXFS Software for Mac OS X

You can modify the behavior of the CXFS client daemon (`cxfs_client`) by placing options in the `/usr/cluster/bin/cxfs_client.options` file. The available options are documented in the `cxfs_client` man page.

Caution: Some of the options are intended to be used internally by SGI only for testing purposes and do not represent supported configurations. Consult your SGI service representative before making any changes.

To see if `cxfs_client` is using the options in `cxfs_client.options`, enter the following:

```
macosx# ps -axwww | grep cxfs
```

For example:

```
macosx# ps -axwww | grep cxfs
611 ??         0:06.17 /usr/cluster/bin/cxfs_client -D trace -z
```

Removing the CXFS Software for Mac OS X

After terminating any applications that access CXFS filesystems on the Mac OS X node, execute the following:

```
macosx# sudo /usr/cluster/bin/uninstall-cxfs
```

Restart the system to unload the CXFS module from the Mac OS X kernel.

Recognizing Storage Changes for Mac OS X

If you make changes to your storage configuration, you may have to reboot your machine because there is currently no mechanism in Mac OS X to reprobe the storage.

Switching Between 64–bit Kernel and 32–bit Kernel on Snow Leopard or Lion,

To determine whether your Snow Leopard, or Lion machine boots with the 32-bit kernel or the 64-bit kernel, you can examine the output from the `system_profiler` command.

For example, the following output indicates 32-bit:

```
snowleopard$ system_profiler | grep -i kernel
      Kernel Version: Darwin 10.8.0
      64-bit Kernel and Extensions: No
```

The following output indicates 64-bit:

```
snowleopard$ system_profiler | grep -i kernel
      Kernel Version: Darwin 10.8.0
      64-bit Kernel and Extensions: Yes
```

To use the System Profiler GUI:

1. Select the following menu:

Apple Menu
> About This Mac
> More Info

2. In the left pane, click **Software**
3. Examine the **64-bit Kernel and Extensions** field:
   - **No** indicates 32-bit
   - **Yes** indicates 64-bit

If you need to switch between 64-bit and 32-bit, see the following website:

http://support.apple.com/kb/HT3773

GRIO on Mac OS X

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Mac OS X platform if GRIO is enabled on the server-capable administration node. Application bandwidth reservations must be explicitly released by the application before exit. If the application terminates unexpectedly or is killed, its bandwidth reservations are not automatically released and will cause a bandwidth leak. If this happens, the lost bandwidth could be recovered by rebooting the client node.

A Mac OS X node can mount a GRIO-managed filesystem and supports node-level reservations. A Mac OS X node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

System Tunable Kernel Parameters on Mac OS X

SGI recommends that you use the same settings for kernel system tunable parameters on all applicable nodes in the cluster.

Caution: Before changing any parameter, you should understand the ramifications of doing so on your system. You should only modify debugging parameters at the recommendation of SGI.

This section discusses the following:

For more information, see the appendix about system tunable parameters in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Making Permanent Parameter Changes on Mac OS X

You can change a parameter permanently across reboots on Mac OS X by adding it to the `/etc/sysctl.conf` file. Use the following format:

prefix`.`*systune*`=`*value*

where:

- *prefix* is one of the following:

  ```
  cxfs.cell
  cxfs.fs
  ```

- *systune* is the parameter name, such as `enable_readdir_type`
- *value* is the value you want to set for the parameter, such as `1`

Note: Do not use spaces around the `=` character.

For example, to permanently set the `enable_readdir_type` parameter (which has the `cxfs.fs` prefix) to `1`, add the following line to the `/etc/sysctl.conf` file:

```
cxfs.fs.enable_readdir_type=1
```

The change will take effect upon reboot.

Making Temporary Parameter Changes on Mac OS X

For a temporary change to a dynamic parameter on a Mac OS X system, use the `sysctl`(8) command:

```
macosx# sysctl -w prefix.systune=value
```

where:

- *prefix* is one of the following:

  ```
  cxfs.cell
  cxfs.fs
  ```

- *systune* is the parameter name, such as `enable_readdir_type`
- *value* is the value you want to set for the parameter, such as `1`

Note: Do not use spaces around the = character.

For example, set temporarily set the `enable_readdir_type` parameter (which has the `cxfs.fs` prefix) to `1`:

```
macosx# sysctl -w cxfs.fs.enable_readdir_type=1
cxfs.fs.enable_readdir_type: 0 -> 1
```

Querying a Current Parameter Setting on Mac OS X

To query the current setting of a parameter on a Mac OS X system, use the `sysctl`(8) command:

```
macosx# sysctl prefix.systune
```

where:

- *prefix* is one of the following:

  ```
  cxfs.cell
  cxfs.fs
  ```

- *systune* is the parameter name, such as `enable_readdir_type`

For example, to query the current setting of the `enable_readdir_type` parameter (which has the `cxfs.fs` prefix):

```
macosx# sysctl cxfs.fs.enable_readdir_type
cxfs.fs.enable_readdir_type: 1
```

Static Site-Configurable Parameters on Mac OS X

Changes to static parameters require a reboot.

Caution: You should only change site-configurable system tunable kernel parameters if you are fully aware of the consequences or if directed to do so by SGI Support.

mtcp_hb_period

The `mtcp_hb_period` parameter specifies (in hundredths of a second) the length of time that CXFS waits for CXFS kernel heartbeat from other nodes before declaring node failure. The same value must be used on all nodes in the cluster.

Range of values:

- Default: 500 (5 seconds) *Recommended*
- Minimum: 100
- Maximum: 12000

Prefix: `cxfs.cell`

Dynamic Parameters for Debugging Purposes Only on Mac OS X

Changes to dynamic parameters take affect immediately.

Caution: You should only modify debugging parameters if directed to do so by SGI Support.

cell_tkm_feature_disable

Disables selected features of the token module by setting a hexadecimal flag bit:

- `0x1` disables speculative token acquisition
- `0x2` (unused)
- `0x4` disables token prefetching
- `0x8` uses multiple RPCs to obtain a token set if the rank and class conflict
- `0x10` disables token lending
- `0x20` disables the blocking of cached tokens
- `0x40` disables range tokens

Range of values:

- Default: `0`
- Maximum: `0`
- Minimum: `0x7fff`

Prefix: `cxfs.cell`

`enable_readdir_type`

The `enable_readdir_type` parameter determines whether the metadata server returns valid information when issuing a `readdir()` on a CXFS filesystem. By default, the returning `dirent.d_type` is set to `DT_UNKNOWN`. However, if an application requires that the `dirent.d_type` value be set to a valid value, you can force the metadata server to return valid information by setting the `enable_readdir_type` parameter on the Mac OS X node to `1`.

Range of values:

- `0` disables (default)
- `1` enables

Prefix: `cxfs.fs`

`large_resourcefork_xa_action`

The `large_resourcefork_xa_action` parameter specifies how files with large resource fork extended attributes (those larger than 64 KB) will be handled on a CXFS filesystem on a Mac OS X node.

Range of values:

- `0` stores all resource fork extended attributes in AppleDouble format (default)
- `1` strips large resource fork extended attributes and stores all other resource fork extended attributes in native format
- `2` returns `E2BIG` for large resource fork extended attributes and stores all other resource fork extended attributes in native format

Prefix: `cxfs.fs`

Note: If you set the `large_resourcefork_xa_action` to `1` or `2`, you should run the `xattr_convert` command with the `-p` option to purge old AppleDouble files on all mounted CXFS filesystems.

Troubleshooting for Mac OS X

This section discusses the following:

Also see:

The cxfs_client Daemon is Not Started on Mac OS X

Confirm that the cxfs_client is not running. The following command would list the cxfs_client process if it were running:

```
macosx# ps -auxww | grep cxfs_client
```

Check the cxfs_client log file for errors.

Restart cxfs_client as described in "Start/Stop cxfs_client for Mac OS X" on page 88 and watch the cxfs_client log file for errors.

XVM Volume Name is Too Long on Mac OS X

On Mac OS X nodes, the following error message in the system.log file indicates that the volume name is too long and must be shortened so that the Mac OS X node can recognize it:

```
devfs: volumename name slot allocation failed (Errno=63)
```

See "Limitations and Considerations for Mac OS X" on page 69.

Large Log Files on Mac OS X

The /var/log/cxfs_client log file may become quite large over a period of time if the verbosity level is increased.

To manually rotate this log file, use the -z option in the /usr/cluster/bin/cxfs_client.options file.

See the cxfs_client.options man page and "Log Files on Mac OS X" on page 68.

Slow Performance on Mac OS X Due to Token Optimizations

Note: You should modify cell_tkm_feature_disable only if directed to do so by SGI Support.

CXFS token prefetch and range tokens are designed as optimizations for applications using CXFS filesystems on a CXFS client. However, under some workloads, token prefetch may actually slow performance and range tokens may cause token hangs. If

directed to do so by SGI Support, you can use the `cell_tkm_feature_disable` system tunable parameter to disable these features on Mac OS X nodes:

- To disable token prefetch:

  ```
  macosx# sysctl -w cxfs.cell.cell_tkm_feature_disable=4
  ```

- To disable range tokens:

  ```
  macosx# sysctl -w cxfs.cell.cell_tkm_feature_disable=64
  ```

- To disable both token prefetch and range tokens:

  ```
  macosx# sysctl -w cxfs.cell.cell_tkm_feature_disable=68
  ```

- To reenable both token prefetch and range tokens (returning to the default behavior):

  ```
  macosx# sysctl -w cxfs.cell.cell_tkm_feature_disable=0
  ```

For more information, see "`cell_tkm_feature_disable`" on page 94. To set the parameter across reboots, see "Making Permanent Parameter Changes on Mac OS X" on page 92.

XVM Failover Problems on Lion, Mountain Lion, Mavericks, and Yosemite Nodes

If there are problems with XVM failover on a Lion, Mountain Lion, Mavericks, or Yosemite node, verify that the `xvmfod` daemon is still running by executing a command such as the following:

```
lion# ps -ef | grep xvmfod
```

If `xvmfod` is not running, you must restart CXFS on the node:

```
lion# service cxfs_client start
```

No WWPNs Detected for Mac OS X

If no WWPNs are detected, the following messages will be logged to the `/var/log/cxfs_client` file:

```
hba_wwpn_list warning: No WWPN found from IO Registry
     cis_get_hbas warning: Not able to find WWN (err=Device not
     configured). Falling back to "/etc/fencing.conf".
```

`cis_config_swports_set error fetching hbas`

If no WWPNs are detected, you can manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `/etc/fencing.conf` file enumerates the WWPNs for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for the HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

If used, `/etc/fencing.conf` must contain a simple list of WWPNs, one per line. You must update it whenever the HBA configuration changes, including the replacement of an HBA.

Do the following:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Follow the Fibre Channel cable on the back of the node to determine the port to which it is connected in the switch. Ports are numbered beginning with 0. (For example, if there are 8 ports, they will be numbered 0 through 7.)
3. Connect to the switch and log in as user `admin`. (On Brocade switches, the password is `password` by default).
4. Execute the `switchshow` command to display the switches and their WWPN numbers. For example:

```
brocade04:admin> switchshow
switchName:     brocade04
switchType:     2.4
switchState:    Online
switchRole:     Principal
switchDomain:   6
switchId:       fffc06
switchWwn:      10:00:00:60:69:12:11:9e
switchBeacon:   OFF
port  0: sw  Online        F-Port  20:00:00:01:73:00:2c:0b
port  1: cu  Online        F-Port  21:00:00:e0:8b:02:36:49
```

```
port  2: cu  Online        F-Port  21:00:00:e0:8b:02:12:49
port  3: sw  Online        F-Port  20:00:00:01:73:00:2d:3e
port  4: cu  Online        F-Port  21:00:00:e0:8b:02:18:96
port  5: cu  Online        F-Port  21:00:00:e0:8b:00:90:8e
port  6: sw  Online        F-Port  20:00:00:01:73:00:3b:5f
port  7: sw  Online        F-Port  20:00:00:01:73:00:33:76
port  8: sw  Online        F-Port  21:00:00:e0:8b:01:d2:57
port  9: sw  Online        F-Port  21:00:00:e0:8b:01:0c:57
port 10: sw  Online        F-Port  20:08:00:a0:b8:0c:13:c9
port 11: sw  Online        F-Port  20:0a:00:a0:b8:0c:04:5a
port 12: sw  Online        F-Port  20:0c:00:a0:b8:0c:24:76
port 13: sw  Online        L-Port  1 public
port 14: sw  No_Light
port 15: cu  Online        F-Port  21:00:00:e0:8b:00:42:d8
```

The WWPN is the hexadecimal string to the right of the port number. For example, the WWPN for port 0 is `2000000173002c0b` (you must remove the colons from the WWPN reported in the `switchshow` output to produce the string to be used in the fencing file).

5. Edit or create `/etc/fencing.conf` and add the WWPN for the port determined in step 2. (Comment lines begin with #.)

   For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

   For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

   ```
   # WWPN of the HBA installed on this system
   #
   2000000173002c0b
   ```

6. To configure fencing, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Reporting Mac OS X Problems

Before reporting a problem to SGI, you should run the `cxfsdump`(8) command on a server-capable administration node. See the information in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Note: You could run cxfsdump on each cluster node manually by executing the `/usr/cluster/bin/cxfsdump -local` command individually on each node, but this is less efficient.

Windows Platforms

CXFS supports a client-only node running the Windows operating systems defined in the CXFS Windows release notes. The information in this chapter applies to all of these versions of Windows unless otherwise noted. To perform the procedures in this chapter, you must be logged in as a user with administrator privileges.

This chapter contains the following sections:

Note: Your **Start** menu may differ from the examples shown in this guide, depending upon your start menu preferences. For example, this guide describes selecting the control panel as follows:

Start
> Control Panel

However, on your system this menu could be as follows:

Start
> Settings
> Control Panel

CXFS on Windows

This section contains the following information about CXFS on Windows:

Requirements for Windows

In addition to the items listed in "Requirements" on page 6, CXFS requires at least the following, as documented in the CXFS Windows release note:

- A supported Windows operating system
- One of the following:
  - An Intel Pentium or compatible processor
  - Xeon family with Intel Extended Memory 64 Technology (EM64T) processor architecture, or AMD Opteron family, AMD Athlon family, or compatible processor
- Minimum RAM requirements (more will improve performance): at least 1 GB of physical RAM
- A minimum of 10 MB of free disk space
- Supported host bus adapter (HBA) and appropriate software

Note: You must install the HBA driver.

CXFS Commands on Windows

The following commands are shipped as part of the CXFS Windows package:

```
%windir%\system32\cxfs_client.exe
%ProgramFiles%\CXFS\cxfs_info.exe
%ProgramFiles%\CXFS\cxfsbmap.exe
%ProgramFiles%\CXFS\cxfscp.exe
%ProgramFiles%\CXFS\cxfsdump.exe
%ProgramFiles%\CXFS\frametest.exe
%ProgramFiles%\CXFS\grioadmin.exe
%ProgramFiles%\CXFS\griomon.exe
%ProgramFiles%\CXFS\grioqos.exe
%ProgramFiles%\CXFS\idbg.exe
%ProgramFiles%\CXFS\xvm.exe
```

Note the following:

- A single `cxfs_client` service and two CXFS filesystem drivers are installed as part of the Windows installation. By default, the `cxfs_client` service is configured to run the CXFS filesystem drivers automatically when the first user logs into the node.
- The command `%ProgramFiles%\CXFS\cxfs_info.exe` displays the current state of the node in the cluster in a graphical user interface. See "Log Files and Cluster Status for Windows" and "Verifying the Cluster Status" on page 205.
- The CXFS software for Windows also includes the `grio2lib` library.
- For information about the GRIO commands, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and "GRIO on Windows" on page 172.
- For information about `frametest`, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Log Files and Cluster Status for Windows

This section discusses the following:

See also "Tuning the Verbosity of CXFS Messages in the System Event Log for Windows" on page 174.

Viewing the Log Files for Windows

The Windows node will log important events in the system event log. You can view these events by selecting the following:

Start
> Control Panel
> Administrative Tools
> Event Viewer

For information about the log files created on server-capable administration nodes, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*. The `cxfs_client` service will also log important information to the following file:

```
%ProgramFiles%\CXFS\log\cxfs_client.log
```

When CXFS is first installed, the log file is automatically rotated when it grows to 10 MB. This is set by the `-z` option in the `cxfs_client` service **Additional arguments** window during installation (see Figure 5-7 on page 139) and may be adjusted by following the steps described in "Modifying the CXFS Software for Windows" on page 168.

Using the CXFS Info Window

You may wish to keep the **CXFS Info** window open to check the cluster status and view the log file. To open this informational window on any Windows system, select the following:

Start
> Programs
> CXFS
> CXFS Info

The top of **CXFS Info** window displays the overall state of the cluster environment:

- Number of stable nodes
- Status of the `cms` cluster membership daemon
- Status of XVM
- Status of filesystems
- Status of the cluster
- Status of the local node

The **CXFS Info** window provides the following tabs:

- **Nodes** displays each node in the cluster, its state, and its cell ID number. For more information, see "Verifying the Cluster Status" on page 205.

  Figure 5-1 shows an example of the **CXFS Info** window **Nodes** tab.

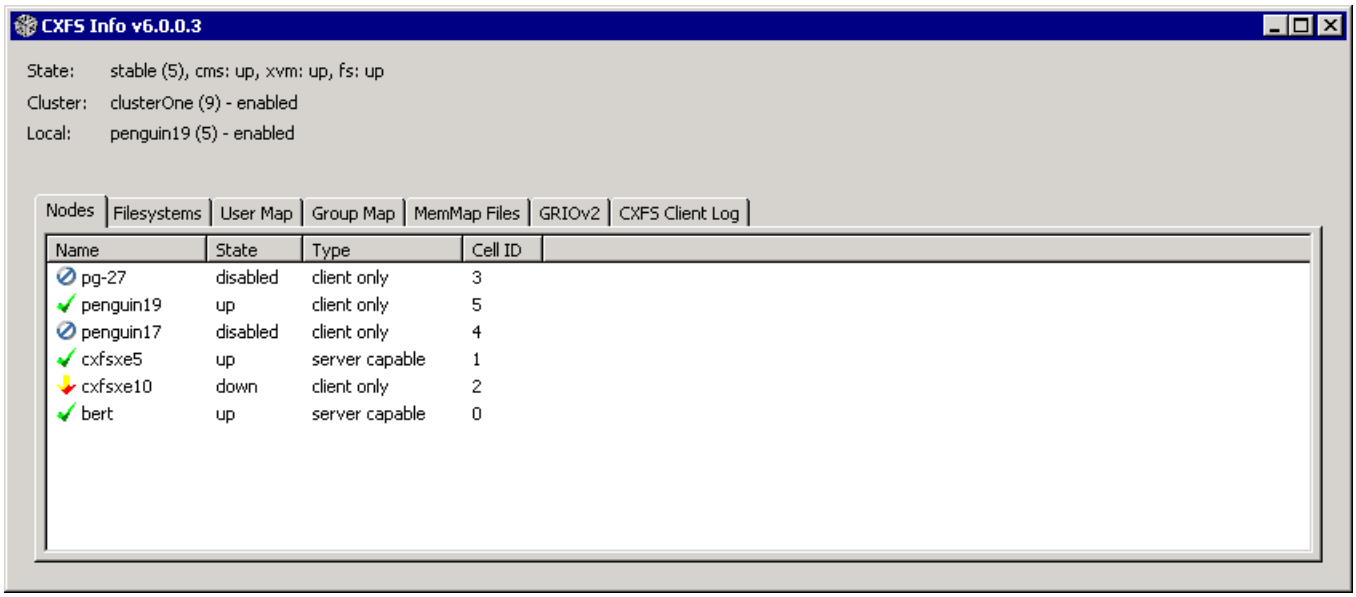

Figure 5-1 CXFS Info Window — Nodes Tab Display

- **Filesystems** displays each CXFS filesystem, its state, size, and other statistics. Figure 5-2 shows an example.

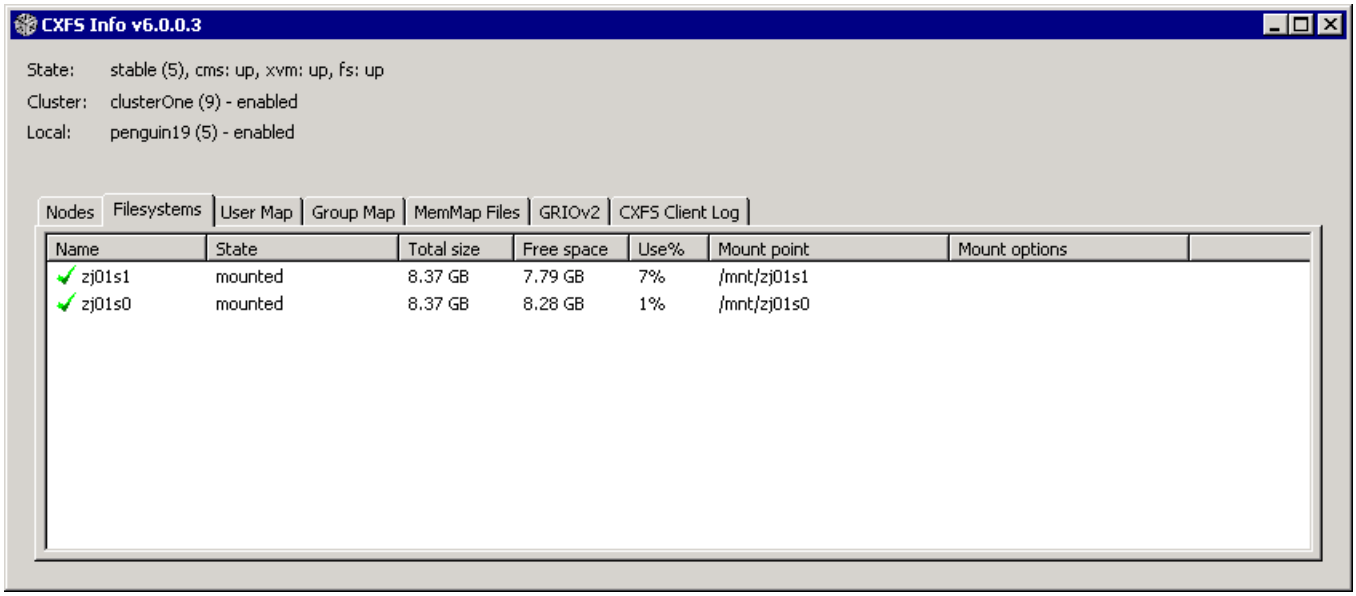

Figure 5-2 CXFS Info Window — Filesystems Tab

- **User Map** displays the usernames that are mapped to UNIX user identifiers. Figure 5-3 shows an example.

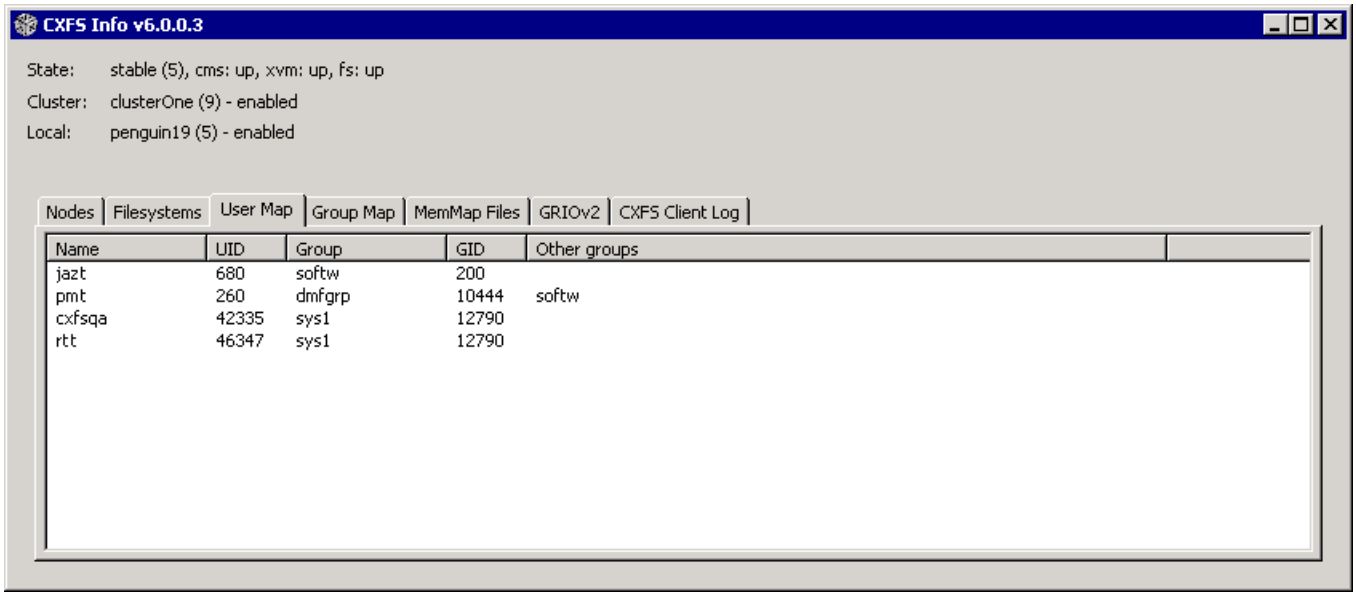

Figure 5-3 CXFS Info Window — User Map Tab

- **Group Map** displays the groups that are mapped to UNIX group identifiers. Figure 5-4 shows an example.

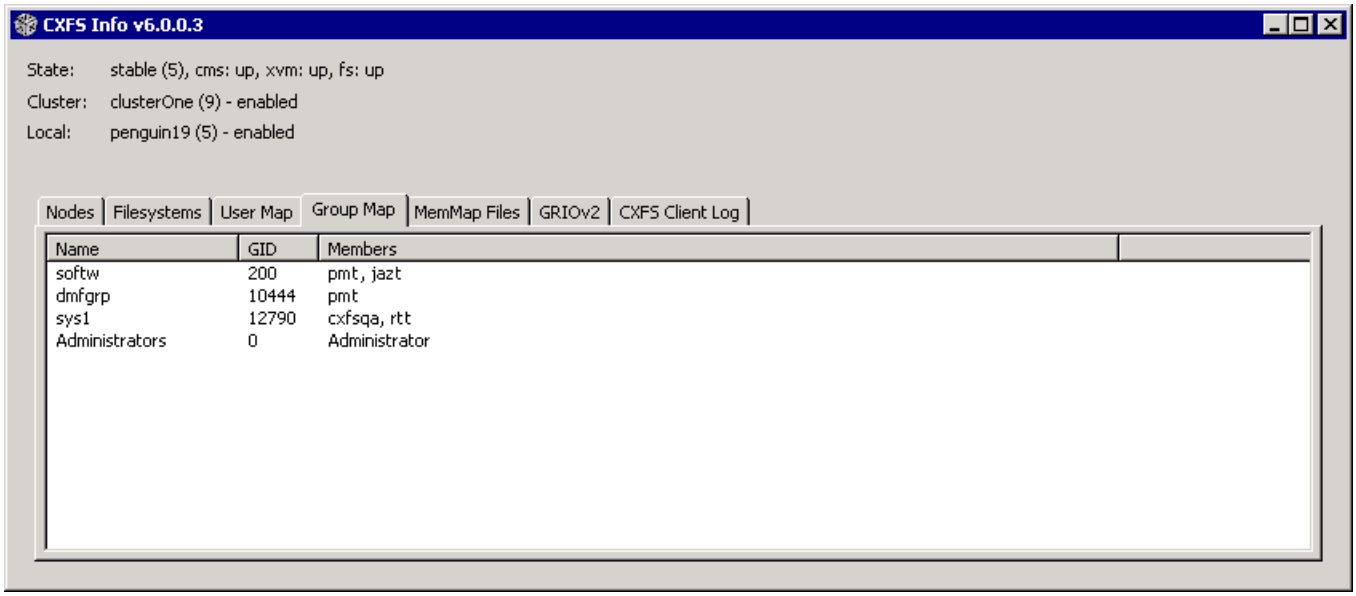

Figure 5-4 CXFS Info Window — Group Map Tab

- **GRIOv2** displays each guaranteed-rate I/O (GRIO) stream, its reservation size, and other statistics. See "GRIO on Windows" on page 172.
- **CXFS Client log** displays the log since the `cxfs_client` service last rebooted. It highlights the text in different colors based on the severity of the output:
  - Red indicates an error, which is a situation that will cause a problem and must be fixed
  - Orange indicates a warning, which is a situation that might cause a problem and should be examined
  - Black indicates general log information that can provide a frame of reference
  - Green indicates good progress in joining membership and mounting filesystems

  Figure 5-5 shows an example.

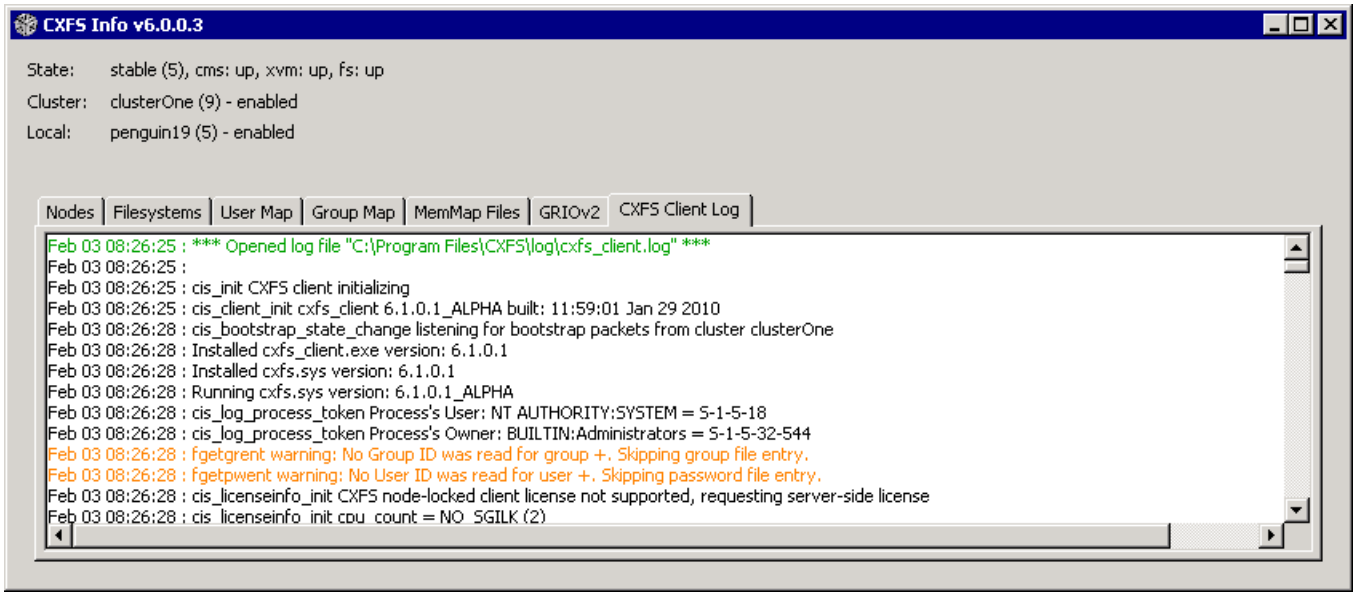

Figure 5-5 CXFS Info Window — CXFS Client Log Tab

The **CXFS Info** icon in the task bar will change from green to yellow or red depending on the state of the node in the cluster:

- Green indicates that the node is in the membership, everything is fully functional, and all enabled filesystems are mounted
- Yellow indicates an in-between state (neither inactive nor stable state)
- Red indicates that CXFS is not running (inactive state)

Also see Figure 5-18 on page 172.

Functional Limitations and Considerations for Windows

This section contains the following:

Warning: DiskManager for Windows Vista, Windows Server 2008, and Windows 7 Destroys Data

After CXFS is installed on the Windows Vista, Windows Server 2008, or Windows 7 platform, you **must not** use DiskManager to format the disks that are exported from the RAID.

Warning: Using DiskManager to format the disks will destroy the XVM labels and therefore the data on the RAID. To make the XVM volumes functional again, you would have to rebuild the XVM labels.

UNIX Perspective of CXFS for Windows

This section describes the differences and limitations of a CXFS filesystem on a Windows node from a UNIX perspective:

- Windows nodes can support multiple CXFS filesystems mounted under a single drive letter. Only one CXFS drive letter may be configured on a Windows node.

  The top-level file structure under the CXFS drive letter consists of an in-memory directory structure that mimics the mount points on the server-capable administration node. The CXFS software creates these directories before mounting the CXFS filesystems. For example, a CXFS filesystem with a mount point of `/mnt/cxfs` on a CXFS Windows node configured to use drive letter `X` will create `X:\mnt\cxfs` during filesystem mount process.

  This file structure supports only creating and deleting directories; there is no support for creating and deleting regular files, renaming directories, and so on. Attempts to perform unsupported actions will generally result in an invalid parameter error. You can perform normal filesystem operations on files and directories beneath the mount points, but an application that must write to the directory directly under the CXFS drive letter will fail.

  Note: A CXFS mount point or directory beneath a mount point can be mapped to another drive letter by using the `subst` command from a command shell to which the application can write. See "Application Cannot Create File Under CXFS Drive Letter" on page 187.

- A Windows node can support regular files, directories, and links. However, it does not support other XFS file types.
- Symbolic links cannot be distinguished from normal files or directories on a Windows node. Opening a symbolic link will open the target of the link, or will report `file not found` if it is a dangling link.
- By default, copying a symbolic link will result in copying the file or directory that the link refers to, rather than the normal UNIX behavior that copies the link itself. To copy the link itself, you must use the `cp -a` option.

For example, on a normal Linux platform:

```
linux# touch file; mkdir dir; ln -sf file file_link; \
ln -sf dir dir_link; cp -a file_link file_link_copy; cp -a dir_link dir_link_copy

linux# file *
dir/:           directory
dir_link:       symbolic link to 'dir'
dir_link_copy:  symbolic link to 'dir'
file:           empty
file_link:      symbolic link to 'file'
file_link_copy: symbolic link to 'file'
```

On a Windows platform using a cygwin shell:

```
user@host /cygdrive/x/mnt/lun0
$ touch file; mkdir dir; ln -sf file file_link; ln -sf dir dir_link; \
cp -a file_link file_link_copy; cp -a dir_link dir_link_copy

user@host /cygdrive/x/mnt/lun0
$ file *
dir:               directory
dir_link.lnk:      symbolic link to 'dir'
dir_link_copy.lnk: MS Windows shortcut
file:              empty
file_link.lnk:     symbolic link to 'file'
file_link_copy.lnk: MS Windows shortcut
```

Windows Perspective of CXFS for Windows

This section describes the differences and limitations of a CXFS filesystem on a Windows node in comparison to other Windows filesystems from a Windows perspective:

- Avoid using duplicate filenames in the same directory that vary only in case. CXFS is case-sensitive, but some Windows applications may not maintain the case of all filenames, which may result in unexpected behavior.
- CXFS software does not export 8.3 alternative filenames. Older Windows applications that only support 8.3 filenames may be unable to open files with longer filenames and may fail with `file not found` errors.
- Avoid using completely uppercase 8.3 filenames. If you use completely uppercase 8.3 filenames, some applications (including Windows Explorer) may incorrectly

assume that only 8.3 filenames are supported by the filesystem and will not preserve case.

- Install the CXFS software components onto a NTFS partition rather than a FAT partition. The security of the following files cannot be guaranteed if these files are installed onto a FAT filesystem:

  ```
  %ProgramFiles%\CXFS\passwd
  %ProgramFiles%\CXFS\group
  ```

- There is no recycle bin; deleted files are permanently deleted.
- There is no automatic notification of directory changes performed by other nodes in the cluster. Applications (such as Windows Explorer) will not automatically update their display if another node adds or removes files from the directory currently displayed.
- A CXFS filesystem cannot be used as the boot partition of a Windows node.
- The volume properties window in Windows Explorer for the CXFS drive letter will display the total capacity of all mounted filesystems and the largest free space on any one of those filesystems.

Forced Unmount on Windows

A forced unmount causes all processes that have open files on the specified filesystem to be unconditionally killed and therefore permit the filesystem to be unmounted without delay. SGI recommends that you enable the forced unmount feature on CXFS filesystems. See:

- "Enable Forced Unmount When Appropriate" on page 16
- "Forcing Unmount of CXFS Filesystems" on page 203

Define LUN 0 on All Storage Devices for Windows XP and Windows Server 2003

Windows XP and Windows Server 2003 (and therefore CXFS) might not detect any LUNs on a storage device if LUN 0 is not defined on the storage device. This problem may occur when **CXFS Info** reports that XVM is up, but one or more filesystems are not mounted and CXFS therefore retries the mount continuously. For more information about this issue, see the following (the problem exists for all supported Windows XP and Windows Server 2003 platforms):

http://support.microsoft.com/kb/821666/en-us

Memory-Mapping Large Files for Windows

You can memory-map a file much larger than 2 GB under Windows, but only up to 2 GB of that file in one or more parts can be mapped into a process at any one time on a 32-bit platform. See the Windows Platform Software Development Kit for more details.

CXFS Mount Scripts for Windows

Windows does not support the CXFS mount scripts.

Norton Ghost Prevents Mounting Filesystems

If Norton Ghost is installed on a node, CXFS cannot mount filesystems on the mount-point driver letter. You must uninstall Norton Ghost in order to use CXFS.

Mapping Network and CXFS Drives

Under Windows XP, users may define their own local set of drive letter mappings that can override the global settings for the host. When identifying the filesystem mapped to a drive letter, Windows XP will check the local mappings and may hide CXFS from the user. Users and administrators of CXFS Windows nodes must avoid mapping network and CXFS drives to the same drive letter.

Windows Filesystem Limitations

A Windows node running CXFS has the following filesystem limitations:

- Does not support shutdown of the `cxfs_client` service via the device manager. If restarting the `cxfs_client` service fails to achieve membership, you must reboot the Windows node.
- Does not support opportunistic locking, also known as *oplocks*. Hosts that are using a CXFS Windows node as an SMB server will not be able to cache data locally. The workaround is to use NFS or Samba to export the filesystem on one of the server-capable administration nodes.
- Enforces the Windows file sharing options when opening a file on the same node, but does not enforce it on other nodes in the cluster.

XFS Filesystem Limitations

Support for unwritten extents is limited on Windows nodes. However, reading and writing unwritten extents will work correctly in the absence of concurrent reading and writing of the same file extent elsewhere in the cluster.

User Account Control for Windows Vista, Windows Server 2008, and Windows 7

By default, User Account Control is enabled for Windows Vista, Windows Server 2008, and Windows 7, but it is not appropriate for use with CXFS. You must therefore disable user account control. See step 4 in "Client Software Installation for Windows" on page 136.

Windows Disks Using DDN RAID

For Windows disks using DDN RAID (versions prior to rm6700), you should set the disk spin-down value so that disks never spin down. (Spinning down a disk could issue a `STOP LUN` command to the storage.)

On Windows XP and Windows Server 2003, do the following:

1. Select the following:

   Start
   > Control Panel
   > Power Options

2. In the **Plugged in** scheme, select **Never** for **Turn off hard disks**.

On Windows Vista and Windows Server 2008, do the following:

1. Select the following:

   Start
   > Control Panel
   > Power Options

2. Select the **High performance** preferred plans.
3. Click the **Change plan settings** link.
4. Click the **Change advanced power settings** link. This will pop-up the **Advanced** settings dialog.

5. Locate the **Hard disk** entry in the tree and expand it.
6. Change the **Turn off hard disk after : Setting: 20 Minutes** setting to **Never**.
7. Click **OK** to save the changes.

On Windows 7, do the following:

1. Select the following:

   Start
   > Control Panel
   > Power Options

2. Click the down-arrow on the right side opposite from **Show additional plans** to reveal the **High performance** plan.
3. Select the **High performance** preferred plans.
4. Click the **Change plan settings** link.
5. Locate the **Hard disk** entry in the tree and expand it.
6. Verify that the setting is **Turn off hard disk after : Setting: Never**.
7. Click **OK** to exit and save any changes.

Windows Time Service Default Synchronization

The Windows Time Service is capable of synchronizing with NTP servers, but the default configuration synchronizes only once a week. SGI recommends modifying the default configuration to keep Windows nodes more closely synchronized. See the Microsoft documentation for the Windows Time Service for details, including the following:

http://technet.microsoft.com/en-us/library/bb490605.aspx

DMF and Memory-Mapped Files on Windows

When Windows has a file memory mapped, applications such as DMF may hang when trying to access the file because there is no way to force Windows release its file mapping. Using memory-mapped files can be unavoidable because a file can become memory-mapped by many different methods, such as by moving the mouse over the file's icon or viewing the folder that contains the file. Windows does not notify the

filesystem that the file is memory-mapped, and Windows will keep a file memory-mapped until it is forced to relinquish the file.

The Windows tuning parameter DisableMemMapCoherency can force Windows to allow the node to return tokens even if the file is memory mapped, but then there is a risk of data corruption. See "Memory-Mapping Coherency for Windows" on page 175.

Performance Considerations for Windows

The following are performance considerations on a CXFS Windows node:

- Using CIFS to share a CXFS filesystem from a CXFS Windows node to another Windows host is not recommended for the following reasons:
  - Metadata operations sent to the Windows node must also be sent to the CXFS metadata server causing additional latency
  - CXFS Windows does not support opportunistic locking, which CIFS uses to improve performance (see "Windows Filesystem Limitations" on page 116)

  For optimal performance, SGI recommends that you use Samba on the CXFS active metadata server to export CXFS filesystems to other nodes that are not running CXFS.
- Windows supplies autonotification APIs for informing applications when files or directories have changed on the local client. With each notification, Windows Explorer will do a full directory lookup. Under CXFS, directory lookups can require multiple RPCs to the server (about 1 per 30 files in the directory), resulting in a linear increase in network traffic. This can grow to megabytes per second for directories with large numbers of files.

  For better performance, do one of the following:
  - Select the destination folder itself
  - Close the drive tree or mount point folder by clicking on the |+| on the drive icon or mount point folder
- If you open the Windows Explorer **Properties** window on a directory, it will attempt to traverse the filesystem in order to count the number and size of all subdirectories and files; this action is the equivalent of running the UNIX du command. This can be an expensive operation, especially if performed on directories between the drive letter and the mount points, because it will traverse all mounted filesystems.

- Virus scanners, Microsoft Find Fast, and similar tools that traverse a filesystem are very expensive on a CXFS filesystem. Such tools should be configured so that they do not automatically traverse the CXFS drive letter.
- The mapping from Windows user and group names to UNIX identifiers occurs as the CXFS software starts up. In a Windows domain environment, this process can take a number of seconds per user for usernames that do not have accounts within the domain. If you are using a `passwd` file for user identification and the file contains a number of unknown users on the Windows node, you should remove users who do not have accounts on the Windows nodes from the `passwd` file that is installed on the Windows nodes.

  This issue has less impact on Windows nodes in a workgroup than on those in a domain because the usernames can be quickly resolved on the node itself, rather than across the network to the domain controller.
- With 1-GB fabric to a single RAID controller, it is possible for one 32-bit 33-MHz QLogic card to reach the bandwidth limitations of the fabric, and therefore there will be no benefit from load balancing two HBAs in the same PCI bus. This can be avoided by using 2-GB fabric and/or multiple RAID controllers.
- For load balancing of two HBAs to be truly beneficial, the host must have at least one of the following three attributes:
  - A 64-bit PCI bus
  - A 66-MHz PCI bus
  - Multiple PCI buses
- Applications running on a Windows node should perform well when their I/O access patterns are similar to those described in the section that discusses when to use CXFS in Chapter 1 of *CXFS 7 Administrator Guide for SGI InfiniteStorage*.
- The maximum I/O size issued by the QLogic HBA to a storage target and the command tag queue length the HBA maintains to each target can be configured in the registry. See "System-Tunable Parameters for Windows" on page 173.

Access Controls for Windows

The XFS filesystem used by CXFS implements and enforces UNIX mode bits and POSIX access control lists (ACLs), which are quite different from Windows file attributes and access control lists. The CXFS software attempts to map Windows

access controls to the UNIX access controls for display and manipulation, but there are a number of features that are not supported (or may result in unexpected behavior) that are described here.

This section contains the following:

User Identification for Windows

The CXFS software supports several user identification mechanisms, which are described in "User Identification Mapping Methods for Windows" on page 122. Only Windows user and group names that exactly match entries in the configured user list will be mapped to those user IDs (UIDs) and group IDs (GIDs). Windows users and groups that do not have a match in the mapping list will be mapped to `nobody`. Users and groups in the mapping list that do not match a Windows user or group are ignored. To avoid confusion and improve performance, you should remove unused users and groups from the mapping list.

The following additional mappings are automatically applied:

- User **Administrator** is mapped to `root` (UID = 0)
- Group **Administrators** is mapped to `sys` (GID = 0)

A user's default UNIX GID is the default GID in the `passwd` listing for the user and is not based on a Windows group mapped to a UNIX group name.

You can display the users and groups that have been successfully mapped by looking at the tables for the **User Map** and **Group Map** tabs in the **CXFS Info** window.

The following sections assume that a CXFS Windows node was configured with the following `passwd` and `group` files:

```
C:\> type %ProgramFiles%\CXFS\passwd
root::0:0:Super-User:/root:/bin/tcsh
guest::998:998:Guest Account:/usr/people/guest:/bin/csh
fred::1040:402:Fred Costello:/users/fred:/bin/tcsh
diane::1052:402:Diane Green:/users/diane:/bin/tcsh

C:\> type %ProgramFiles%\CXFS\group
sys::0:root,bin,sys,adm
root::0:root
guest:*:998:
video::402:fred,diane
audio::403:fred
```

User Identification Mapping Methods for Windows

User identification can be performed by choosing one of the following methods for the **User ID mapping lookup sequence** item of the **Enter CXFS Details** window:

- **files:** `/etc/passwd` and `/etc/group` files from the metadata server copied onto the clients. The format of the `passwd` and `group` files for CXFS Windows is the same as on the metadata server. In the `passwd` file, only the user name, `uid` and `gid` fields are used. In the `group` file, only the group name, `gid`, and member list are used. Other fields may be removed to make the file more readable. If you select this method, you must install the `passwd` and `group` files immediately after installing the CXFS software, as described in "Performing User Configuration for Windows" on page 145.
- **ldap_activedir:** Windows Active Directory server with Services for UNIX (SFU) installed, which uses lightweight directory access protocol (LDAP).

  The **ldap_activedir** method configures the CXFS Windows software to communicate with the Active Directory for the CXFS node's domain. With the Windows Services for UNIX (SFU) extensions, the Active Directory User Manager lets you define UNIX identifiers for each user and export these identifiers as an LDAP database.

Permissions on the Active Directory server must allow Authenticated Users to read the SFU attributes from the server. Depending on the installation and configuration of the server, LDAP clients may or may not be able to access the SFU attributes. For more information, see"cxfs_client Service Cannot Map Users other than `Administrator` for Windows" on page 185.

This configuration requires a domain controller that is installed with the following:

– Windows Server 2003 with Active Directory.

– Windows Services for UNIX (SFU) version 2 or later with the NFS server component installed. SGI recommends SFU version 3.5.

Note: The domain controller does not have to be a CXFS node.

- **ldap_generic:** Generic LDAP lookup for UNIX users and groups from another LDAP server.

  The **ldap_generic** method configures the CXFS software to communicate with an LDAP database that maps user names and group names to UNIX identifiers.

  Following is an example of a user record:

```
# ldap2, people, example.com
dn: uid=ldap2,ou=people,dc=example,dc=com
cn: Ldap Tu User
givenName: Ldap
homeDirectory: /home/ldap2
loginShell: /bin/bash
objectClass: top
objectClass: posixAccount
objectClass: inetOrgPerson
sn: User
uid: ldap2
uidNumber: 1102
gidNumber: 1100
```

Following is an example of a group record:

```
# ldapgroup, group, example.com
dn: cn=ldapgroup,ou=group,dc=example,dc=com
cn: ldapgroup
gidNumber: 1100
memberUid: ldap1,ldap2
objectClass: top
objectClass: posixGroup
objectClass: groupOfNames
```

Note: For the group mapping, you must use `memberUid`, not `member`. You should also use the simple `uid` (such as `myname`) rather than Descriptive Notation (`uid=myname,ou=people,dc=mycompany,dc=com`).

For an example of the window, see Figure 5-7 on page 139.

You must select one of these as the primary mapping method during installation, but you can change the method at a later time, as described in "Modifying the CXFS Software for Windows" on page 168.

Optionally, you can select a secondary mapping method that will be applied to users that are not covered by the first method. If you choose a primary and a secondary mapping method, one of them must be **files**.

For example, suppose the user has selected **ldap_generic** as the primary method and **files** as the secondary method. A user mapping will be created for all suitable **ldap_generic** users and this mapping will be extended with any additional users found in the secondary method (**files**). The primary method will be used to resolve any duplicate entries.

Suppose the primary method (**ldap_generic**) has users for UIDs 1, 2 and 3, and the secondary method (**files**) has users for UIDs 2 and 4. The username for UIDs 1, 2 and 3 will be determined by the **ldap_generic** method and the username for UID 4 will be determined by the **files** method. If the LDAP lookup failed (such as if the LDAP server was down), a user mapping for UIDs 2 and 4 would be generated using the **files** method.

The default behavior is to use the **files** method to map Windows usernames to UNIX UIDs and GIDs, with no secondary method selected.

Regardless of the method used, the consistent mapping of usernames is a requirement to ensure consistent behavior on all CXFS nodes. Most platforms can be configured to use an LDAP database for user identification.

Matching Windows Users and Groups with CXFS Users and Groups

If a file (or a component of the path to the file) has an owner or group that does not exist on the Windows node, Windows may assume that there is a significant security vulnerability and may not allow access to the file or path. This may be true even if the file and every component of the path is world readable/writable.

To avoid this problem, do the following:

1. Create Windows Users and Groups for every user and group likely to be found on the CXFS filesystems.
2. Configure CXFS user and group mapping so that the above Windows Users and Groups are mapped to the CXFS users and groups.

Enforcing Access to Files and Directories for Windows

Access controls are enforced on the CXFS metadata server by using the mapped UID and GID of the user attempting to access the file. Therefore, a user can expect the same access on a Windows node as any other node in the cluster when mounting a given filesystem. Access is determined using the file's ACL (if one is defined) or by using the file's mode bits.

ACLs that are set on any files or directories are also enforced as they would be on any Linux node. The presentation of ACLs is customized to the interfaces of Windows Explorer, so the enforcement of the ACL may vary from an NTFS ACL that is presented in the same way. A new file will inherit the parent directory default ACL, if one is defined.

The user `Administrator` has read and write access to all files on a CXFS filesystem, in the same way that `root` has superuser privileges on a UNIX node.

The following example is a directory listing on the metadata server:

```
MDS# ls -l
drwxr-x---    2 fred   video           6 Nov 20 13:33 dir1
-rw-r-----    1 fred   audio           0 Nov 20 12:59 file1
-rw-rw-r--    1 fred   video           0 Nov 20 12:59 file2
```

Users will have the following access to the contents of this directory:

- `dir1` will be readable, writable, and searchable by user `fred` and `Administrator`. It will be readable and searchable by other users in group `video`, and not accessible by all other users.
- `file1` will be readable and writable to user `fred` and `Administrator` on a CXFS Windows node. It can also be read by other users in group `audio`. No other users, including `diane` and `guest`, will be able to access this file.
- `file2` will be readable by all users, and writable by user `fred`, `diane` (because she is in group `video`), and `Administrator`.

Viewing and Changing File Attributes with Windows Explorer

File permissions may be viewed and manipulated in two different ways when using Windows Explorer:

- By displaying the list of attributes in a detailed directory listing; this is the most limited approach
- By selecting properties on a file

The only file attribute that is supported by CXFS is the read-only attribute; other attributes will not be set by CXFS and changes to those attributes will be ignored.

If the user is not permitted to write to the file, the read-only attribute will be set. The owner of the file may change this attribute and modify the mode bits. Other users, including the user `Administrator`, will receive an error message if they attempt to change this attribute.

Marking a file read-only will remove the write bit from the user, group, and other mode bits on the file. Unsetting the read-only attribute will make the file writable by the owner only.

For example, selecting file properties on `file1` using Windows Explorer on a CXFS Windows node will display the read-only attribute unset if logged in as `Administrator` or `fred`, and it will be set for `diane` and `guest`.

Only user `fred` will be able to change the attribute on these files, which will change the files under UNIX to the following:

```
-r--r-----    1 fred   audio           0 Nov 20 12:59 file1
-r--r--r--    1 fred   video           0 Nov 20 12:59 file2
```

If `fred` then unset these flags, only he could write to both files:

```
-rw-r-----    1 fred   audio           0 Nov 20 12:59 file1
-rw-r--r--    1 fred   video           0 Nov 20 12:59 file2
```

Viewing and Changing File Permissions with Windows Explorer

By selecting the **Security** tab in the **File Properties** window of a file, a user may view and change a file's permissions with a high level of granularity.

Windows Explorer will list the permissions of the file's owner and the file's group. The `Everyone` group, which represents the mode bits for other users, will also be displayed if other users have any access to the file. Not all Windows permission flags are supported.

The permissions on `file1` are displayed as follows:

```
audio (cxfs1\audio)           Allow: Read
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

Using the **Advanced** button, `file1` is displayed as follows:

```
Allow    Fred Costello (cxfs1\fred)    Special
Allow    audio (cxfs1\audio)           Read
```

User `fred` is listed as having `Special` access because the permission flags in the next example do not exactly match the standard Windows permissions for read and write access to a file. Select `Fred Costello` and then click **View/Edit** to display the permission flags listed in Table 5-1. (The table displays the permissions in the order in which they appear in the **View/Edit** window). You can choose to allow or deny each flag, but some flags will be ignored as described in Table 5-1.

Table 5-1 Permission Flags that May Be Edited

Permission	Description
`Traverse Folder / Execute File`	Used to display and change the execute mode bit on the file or directory
`List Folder / Read Data`	Used to display and change the read mode bit on the file or directory
`Read Attributes`	Set if the read mode bit is set; changing this flag has no effect
`Read Extended Attributes`	Set if the read mode bit is set; changing this flag has no effect
`Create Files / Write Data`	Used to display and change the write mode bit on the file or directory
`Create Folders / Append Data`	Set if the write mode bit is set; changing this flag has no effect
`Write Attributes`	Set if the write mode bit is set; changing this flag has no effect
`Write Extended Attributes`	Set if the write mode bit is set; changing this flag has no effect
`Delete Subfolders and Files`	Set for directories if you have write and execute permission on the directory; changing this flag has no effect
`Delete`	Never set (because delete depends on the parent directory permissions); changing the flag has no effect
`Read Permissions`	Always set; changing the flag has no effect
`Change Permissions`	Always set for the owner of the file and the user `Administrator`; changing this flag has no effect
`Take Ownership`	Always set for the owner of the file and the user `Administrator`; changing this flag has no effect

The permissions for `file2` are displayed as follows:

```
Everyone                      Allow: Read
video (cxfs1\video)           Allow: Read, Write
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

The permissions for `dir1` are displayed as follows:

```
Fred Costello (cxfs1\fred)    Allow:
Video (cxfs1\video)           Allow:
```

Note: In this example, the permission flags for directories do not match any of the standard permission sets, therefore no `Allow` flags are set.

In general, you must click the **Advanced** button to see the actual permissions of directories. For example:

```
Allow   Fred Costello   Special                This folder only
Allow   video           Read & Execute         This folder only
```

The `dir1` directory does not have a default ACL, so none of these permissions are inherited, as indicated by the `This folder only` tag, when a new subdirectory or file is created.

Viewing and Changing File Access Control Lists (ACLs) for Windows

If the file or directory has an ACL, the list may include other users and groups, and the `CXFS ACL Mask` group that represents the Linux ACL mask. See the `chacl`(1)man page on the server-capable administration node for an explanation of Linux ACLs and the mask bits. The effective permissions of all entries except for the owner will be the intersection of the listed permissions for that user or group and the mask permissions. Therefore, changing the `CXFS ACL Mask` permissions will set the maximum permissions that other listed users and groups may have. Their access may be further constrained in the specific entries for those users and groups.

By default, files and directories do not have an ACL, only mode bits, but an ACL will be created if changes to the permissions require an ACL to be defined. For example, granting or denying permissions to another user or group will force an ACL to be created. Once an ACL has been created for a file, the file will continue to have an ACL even if the permissions are reduced back to only the owner or group of the file. The `chacl`(1) command under Linux can be used to remove an ACL from a file.

For example, `fred` grants `diane` read access to `file1` by adding user `diane` using the file properties dialogs, and then deselecting `Read & Execute` so that only `Read` is selected. The access list now appears as follows:

```
audio (cxfs1\audio)           Allow: Read
Diane Green (cxfs1\diane)     Allow: Read
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

After clicking **OK**, the properties for `file1` will also include the `CXFS ACL Mask` displayed as follows:

```
audio (cxfs1\audio)           Allow: Read
CXFS ACL Mask (cxfs1\CXFS...) Allow: Read
Diane Green (cxfs1\diane)     Allow: Read
Fred Costello (cxfs1\fred)    Allow: Read, Write
```

Note: You should select and deselect entries in the `Allow` column only, because UNIX ACLs do not have the concept of `Deny`. Using the `Deny` column will result in an ACL that allows everything that is not denied, even if it is not specifically selected in the `Allow` column, which is usually not what the user intended.

Effective Access for Windows

The effective access of user `diane` and group `audio` is read-only. Granting write access to user `diane` as in the following example does not give `diane` write access because the mask remains read-only. However, because user `fred` is the owner of the file, the mask does not apply to his access to `file1`.

For example:

```
audio (cxfs1\audio)                 Allow: Read
CXFS ACL Mask (cxfs1\CXFS...)       Allow: Read
Diane Green (cxfs1\diane)           Allow: Read, Write
Fred Costello (cxfs1\fred)          Allow: Read, Write
```

Restrictions with file ACLs for Windows

If the users and groups listed in a file's permissions (whether mode bits and/or ACL entries) cannot be mapped to users and groups on the Windows node, attempts to display the file permissions in a file properties window will fail with an unknown user or group error. This prevents the display of an incomplete view, which could be misleading.

The owner of the file and users with administrator privileges may change the permissions of a file or directory using Windows Explorer. All other users will get a `permission denied` error message.

Note: A user must use a node that is **not** running Windows to change the ownership of a file because a Windows user takes ownership of a file with Windows Explorer, rather than the owner giving ownership to another user (which is supported by the UNIX access controls).

Inheritance and Default ACLs for Windows

When a new file or directory is created, normally the mode bits are set using a umask of `022`. Therefore, a new file has a mode of `644` and a new directory of `755`, which means that only the user has write access to the file or directory.

You can change this umask during CXFS installation or later by modifying the installation. For more information, see "Client Software Installation for Windows" on page 136 and "Inheritance and Default ACLs for Windows" on page 131.

The four umask options available during installation or modification correspond to the following umask values:

`000`	Everyone can write
`002`	User and group can write
`022`	User only can write (default)
`222`	Read only (no one can write)

Therefore, creating a file on a UNIX CXFS client results in a mode of `644` for a umask of `022`:

```
admin% ls -lda .
drwxr-xr-x    3 fred     video           41 Nov 21 18:01 ./

admin% umask
0022

admin% touch file3
admin% ls -l file3
-rw-r--r--    1 fred     video            0 Nov 21 18:23 file3
```

For more information, see the `umask` man page on the server-capable administration node.

Creating a file in Windows Explorer on a Windows node will have the same result.

A Linux directory ACL may include a default ACL that is inherited by new files and directories, instead of applying the umask. Default ACLs are displayed in the Windows Explorer file permission window if they have been set on a directory. Unlike a Windows inheritable ACL on an NTFS filesystem, a Linux default ACL applies to both new files and subdirectories; there is no support for an inheritable ACL for new files and another ACL for new subdirectories.

The following example applies an ACL and a default ACL to `dir1` and then creates a file and a directory in `dir1`:

```
admin% chacl -b "u::rwx,g::r-x,u:diane:r-x,o::---,m::r-x" \
           "u::rwx,g::r-x,u:diane:rwx,o::---,m::rwx" dir1
admin% touch dir1/newfile
admin% mkdir dir1/newdir
admin% ls -D dir1
newdir [u::rwx,g::r-x,u:diane:rwx,o::---,m::r-x/
        u::rwx,g::r-x,u:diane:rwx,o::---,m::rwx]
newfile [u::rw-,g::r-x,u:diane:rwx,o::---,m::r--]
```

The permissions for `dir1` will be as follows:

```
CXFS ACL Mask (cxfs1\CXFS...)  Allow:
Diane Green (cxfs1\diane)      Allow:
Fred Costello (cxfs1\fred)     Allow: Read & Exec, List, Read, Write
Video (cxfs1\video)            Allow: Read & Exec, List, Read
```

After clicking on **Advanced**, the permissions displayed are as follows:,

```
Allow  Fred Costello   Special               This folder, subfolders and files
Allow  video           Read & Execute        This folder, subfolders and files
Allow  Diane Green     Read, Write & Exec    Subfolders and files
Allow  CXFS ACL Mask   Read, Write & Exec    Subfolders and files
Allow  Diane Green     Read & Exec           This folder only
Allow  CXFS ACL Mask   Read & Exec           This folder only
```

If an ACL entry is the same in the default ACL, a single entry is generated for the `This folder, subfolders and files` entry. Any entries that are different will have both `Subfolders and files` and `This folder only` entries.

Adding the first inheritable entry to a directory will cause CXFS to generate any missing ACL entries like the owner, group, and other users. The mode bits for these entries will be generated from the umask.

Adding different `Subfolders Only` and `Files Only` entries will result in only the first entry being used because a Linux ACL cannot differentiate between the two.

HBA Installation for Windows

Note: SGI recommends that you use XVM failover V2 and disable any failover capability provided by Windows or the HBA. See "Configuring the `failover2.conf` File for Windows" on page 146.

The Fibre Channel HBA should be installed according to the HBA vendor's hardware and driver installation instructions.

For information regarding large logical unit (LUN) support under Windows, see the HBA vendor's documentation and Microsoft's support database:

http://support.microsoft.com/default.aspx?scid=kb;en-us;Q310072
http://support.microsoft.com/default.aspx?scid=kb;en-us;Q245637

To confirm that the HBA and driver are correctly installed, select the following to display all of the LUNs visible to the HBA and listed within the Device Manager:

Start
> Control Panel
> Administrative Tools
> Computer Management
> Device Manager
> View
> Devices by connection

The Windows Device Manager hardware tree will differ from one configuration to another, so the actual location of the HBA within the Device Manager may differ. After it is located, any LUNs attached will be listed beneath it.

Preinstallation Steps for Windows

This section provides an overview of the steps that you or a qualified Windows service representative will perform on your Windows nodes prior to installing the CXFS software. It contains the following:

Adding a Private Network for Windows

A private network is required for use with CXFS. See "Use a Private Network" on page 13.

Verifying the Private and Public Networks for Windows

You can confirm that the previous procedures to add private networks were performed correctly by using the `ipconfig` command in a DOS command shell.

Create a DOS command shell with the following sequence:

Start
> Programs
> Accessories
> Command Prompt

In the following example, the 10 network is the private network and the 192.168.0 network is the public network on a Windows system:

```
C:\> ipconfig /all
Windows IP Configuration

        Host Name . . . . . . . . . . . . : cxfs1
        Primary Dns Suffix  . . . . . . . : cxfs-domain.sgi.com
        Node Type . . . . . . . . . . . . : Unknown
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No
```

```
        DNS Suffix Search List. . . . . : cxfs-domain.sgi.com
                                          sgi.com

Ethernet adapter Public:

        Connection-specific DNS Suffix  . : cxfs-domain.sgi.com
        Description . . . . . . . . . . . : 3Com EtherLink PCI
        Physical Address. . . . . . . . . : 00-01-03-46-2E-09
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.0.101
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1
        DNS Servers . . . . . . . . . . . : 192.168.0.x

Ethernet adapter Private:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : 3Com EtherLink PCI
        Physical Address. . . . . . . . . : 00-B0-D0-31-22-7C
        Dhcp Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 10.0.0.101
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :
```

Configuring the Windows Firewall for Windows

The Windows firewall will prevent a CXFS Windows node from achieving membership unless several ports are opened using the following applet:

Start
> Control Panel
> Windows Firewall

In the **Exceptions** tab, add the following **Ports**:

UDP on port 5449
TCP on port 5450
TCP on port 5451
UDP on port 5453

Preallocating Space for Directories when Appropriate

Windows copy, Windows Explorer drag-and-drop, and other applications that use small I/O have a high ratio of metadata transfer and can cause the filesystem to become fragmented quickly. Heavy I/O traffic and fragmentation can degrade performance.

To improve the performance of these applications, you can use the `xfs_io extsize` command after you create a new directory that will contain files written with small I/O; the `extsize` command causes the filesystem to preallocate the specified disk space when writing files.

For more information, see the section about preallocating space for directories in the "CXFS Best Practices" chapter of the *CXFS 7 Administrator Guide for SGI InfiniteStorage* and the `xfs_io`(8) man page.

Client Software Installation for Windows

Note: This procedure assumes that the CXFS software is installed under the default path `%ProgramFiles%\CXFS`. If a different path is selected, then that path should be used in its place in the following instructions.

To install the CXFS client software on a Windows node, do the following:

1. Read the *SGI InfiniteStorage Software Platform* release notes CXFS release notes in the `/docs` directory on the ISSP DVD and any late-breaking caveats on the download page.
2. Log onto the Windows node as a user with administrator privileges.
3. Verify that the node has been updated to the correct service pack:

   Start
   > Programs
   > Accessories
   > System Tools
   > System Information

   Note: If you must reinstall the operating system, disconnect the system from the fabric first.

4. *(Windows Vista and later)* Disable User Account Control (requires administrator privileges). By default, User Account Control is enabled for Windows Vista and later, but it is not appropriate for use with CXFS. Do the following to disable it, according to OS type:

   - Windows Vista or Windows Server 2008:

     a. Using the **User Accounts** control panel, click the **Turn User Account Control on or off** link.

     b. Uncheck the **Use User Account Control (UAC) to help protect your computer** check box. Press the **OK** button to confirm your selection.

     c. Reboot the system to apply the changes.

   - Other Windows operating systems:

     a. Using the **User Accounts** control panel, select **Change User Account Control Settings**.

     b. In the **User Account Control Settings** window, move the slider to the bottom **Never notify** setting. Click **OK**.

     c. Reboot the system to apply the changes

5. Transfer the client software that was downloaded onto a server-capable administration node during its installation procedure using `ftp`, `rcp`, or `scp`. The location of the Windows installation program on the server will be as follows:

```
/usr/cluster/client-dist/CXFS_VERSION/windows/all/noarch/setup.exe
```

6. Double-click the **setup.exe** installation program to execute it.

7. Acknowledge the software license agreement when prompted and read the CXFS Windows release notes, which may contain corrections to this guide.

8. Install the CXFS software, as shown in Figure 5-6. If the software is to be installed in a nondefault directory, click **Browse** to select another directory. Click **Next** when finished.

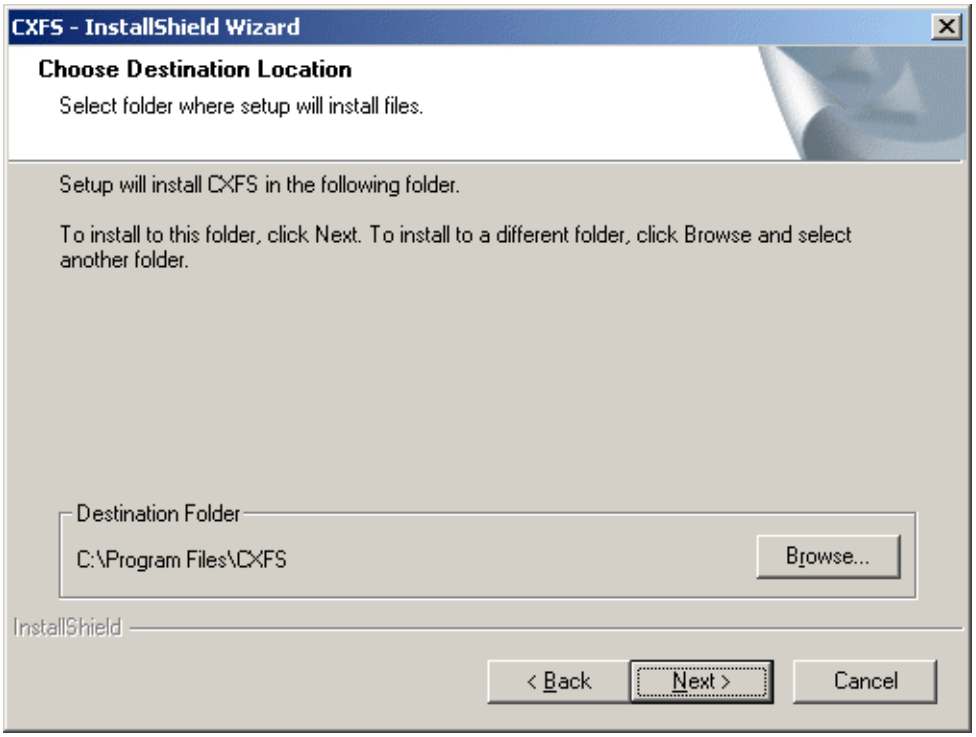

Figure 5-6 Choose Destination Location

9. Enter details for the following fields as shown in Figure 5-7 and click **Next** when finished:

   - **Drive letter for CXFS:** specify the drive letter under which all CXFS filesystems will be mounted. You cannot select a drive letter that is currently in use.
   - **Default Umask:** choose the default umask. For more information on the umask, see "Inheritance and Default ACLs for Windows" on page 131.
   - **User ID mapping lookup sequence:** choose the appropriate primary and (optionally) secondary method. See "User Identification Mapping Methods for Windows" on page 122.
   - **Location of fencing, UNIX /etc/passwd and /etc/group files:** specify the path where the configuration files will be installed and accessed by the CXFS

software if required. The default is the same location as the software under `%ProgramFiles%\CXFS`.

- **IP address of the heartbeat network adapter**: specify the IP address of the private network adapter on the Windows node.
- **Additional arguments**: contains parameters that are used by the `cxfs_client` service when it starts up. For most configurations, this should be left alone. To get a list of options, type `cxfs_client -h` in a command shell (`cmd`) window.

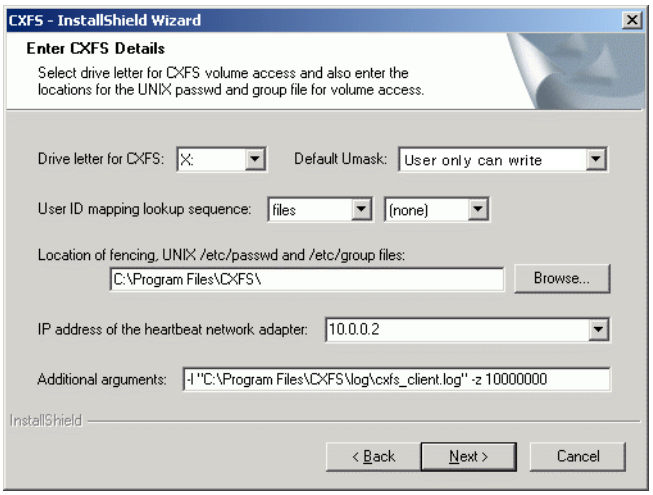

Figure 5-7 Enter CXFS Details

10. If you select **ldap_activedir** as the user ID mapping method, the dialog in Figure 5-8 is displayed after you click **Next**.

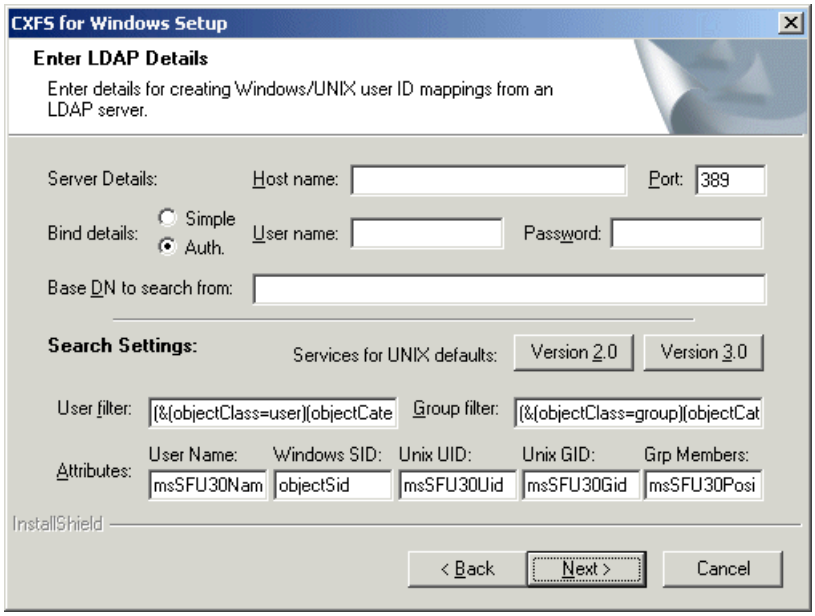

Figure 5-8 Active Directory Details

If you have a standard Active Directory configuration with Windows Services for UNIX (SFU), you need only to select the version of SFU and **Auth** (authenticated) for **Bind details**; doing so will then define the correct Active Directory defaults. The other server details can normally remain blank.

11. If you select **ldap_generic** as the user ID mapping method, the dialog in Figure 5-9 is displayed after you click **Next**. You must provide entries for the **Host name** and the **Base DN to search from** fields. For a standard OpenLDAP server, you can select a simple anonymous bind (default settings with the **User name** and **Password** fields left blank) and select the standard search settings by clicking **Posix**.

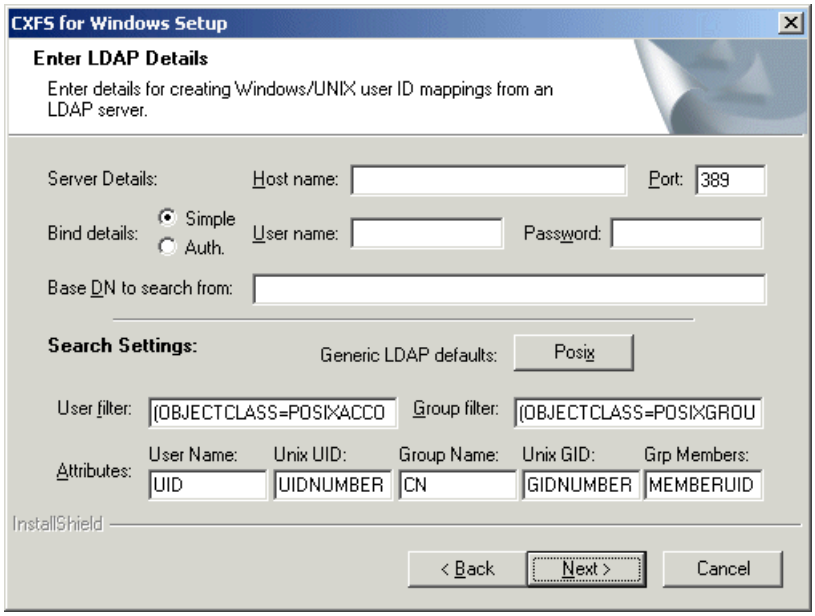

Figure 5-9 Generic LDAP Details

12. Review the settings, as shown in Figure 5-10. If they appear as you intended, click **Next**. If you need to make corrections, click **Back**.

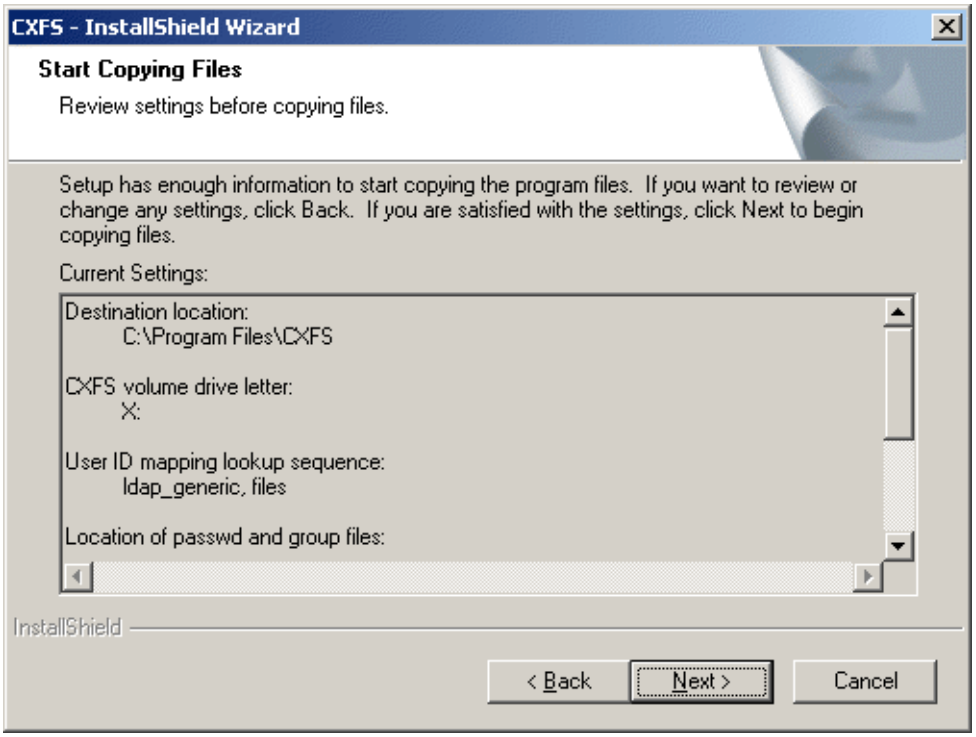

Figure 5-10 Review the Settings

After you click **Next**, the CXFS software will be installed.

13. You will be given the option to start the driver at system start-up, as shown in Figure 5-11. By checking the boxes, you will start the driver automatically when the system starts up and invoke the **CXFS Info** window minimized to an icon. If you choose not to start the `cxfs_client` service automatically, you must start it manually through the **Services** control panel before you can access CXFS filesystems. To manually start the **CXFS Info** window, select the following:

    Start
    > Programs
    > CXFS
    > CXFS Info

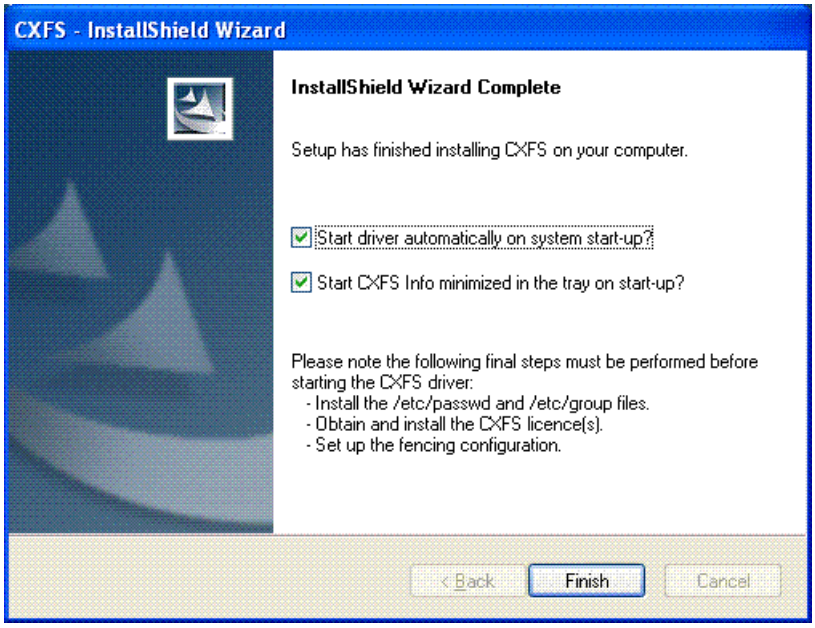

Figure 5-11 Start Driver for the `cxfs_client` Service

14. Choose to restart your computer later if you need to install `passwd` and `group` files or set up fencing; see "Postinstallation Steps for Windows" on page 144. Otherwise, choose to restart your computer now. The default is to restart later, as shown in Figure 5-12. (CXFS will not run until a restart has occurred.)

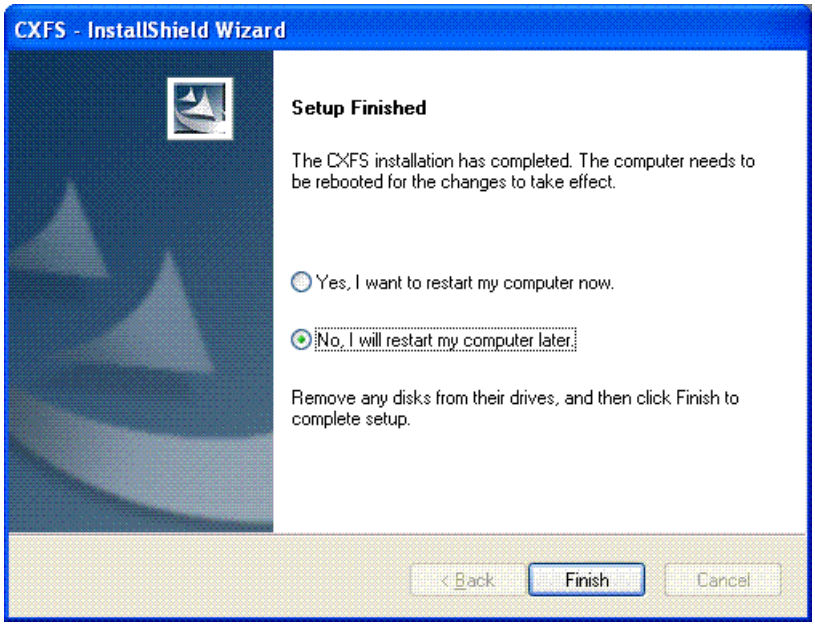

Figure 5-12 Restart the System

Postinstallation Steps for Windows

This section discusses the configuration steps that you should perform after installing CXFS software but before rebooting a Windows node.

The following postinstallation steps are required to ensure the correct operation of the CXFS software:

- "Checking Permissions on the Password and Group Files for Windows" on page 145
- "Performing User Configuration for Windows" on page 145
- "Configuring the `failover2.conf` File for Windows" on page 146
- "Converting an Existing `failover2.conf` File for Windows with FC RAID with RAID" on page 163
- "Configuring I/O Fencing for Windows (FC Only)" on page 163

• "Mapping XVM Volumes to Storage Targets on Windows" on page 163

Checking Permissions on the Password and Group Files for Windows

The permissions on the `passwd` and `group` files must restrict access so that only the system administrator can modify these files. This can be done by right-clicking on the filenames in Windows Explorer and selecting the following:

Properties
> Security

Verify that the permissions are `Read` for `Everyone` and `Full Control` for `Administrators`.

Caution: Failure to set permissions on the `passwd` and `group` files would allow users to change their UID/GID at will and even gain superuser access to the files on the CXFS filesystem.

Performing User Configuration for Windows

If the user mapping is not correctly configured, all filesystem operations will be as user `nobody`.

If you selected the **passwd and group files** user ID mapping method, you must install the `passwd` and `group` files. The default `passwd` and group files that are installed are invalid files containing comments; these invalid files will cause the `cxfs_client` service to generate warnings in its log file and users may not be correctly configured. You must remove the comments in these files when you install the `passwd` and `group` files.

After installing the CXFS software onto the Windows node but before rebooting it, you must install the `/etc/passwd` and `/etc/group` files from the metadata server to the location on the Windows node specified during installation.

The defaults are as follows:

- `/etc/passwd` as `%ProgramFiles%\CXFS\passwd`
- `/etc/group` as `%ProgramFiles%\CXFS\group`

Do the following:

1. Verify that permissions are set as described in "Checking Permissions on the Password and Group Files for Windows" on page 145.
2. If you selected the **Active Directory** method, you must specify the UNIX identifiers for all users of the CXFS node. On the domain controller, run the following to specify the UNIX UID and GID of a given user:

   Start
   > Program Files
   > Administrative Tools
   > Active Directory Users and Computers
   > Users

3. Select a user and then select:

   Properties
   > UNIX Attributes

   The CXFS software will check for changes to the LDAP database every 5 minutes.
4. After the CXFS software has started, you can use **CXFS Info** to confirm the user configuration, regardless of the user ID mapping method chosen. See "User Identification for Windows" on page 121.

If only the `Administrator` user is mapped, see "cxfs_client Service Cannot Map Users other than `Administrator` for Windows" on page 185.

Configuring the `failover2.conf` File for Windows

This section discusses the following methods for configuring the `%ProgramFiles%\CXFS\failover2.conf` file for Windows, depending upon the RAID type (of which FC RAID will support persistent device names in XVM):

- "FC RAID (Persistent XVM Device Names using WWPNs)" on page 147
- "Specific RAID (Nonpersistent XVM Device Names)" on page 149
- "Other RAID (Nonpersistent XVM Device Names)" on page 158

FC RAID (Persistent XVM Device Names using WWPNs)

Note: This procedure applies only to Fibre Channel RAID, for which XVM can provide persistent device names consisting of the World Wide Port Names (WWPNs) of the HBA and RAID controller and the LUN number.

To configure the `failover2.conf` file for Fibre Channel RAID, do the following:

1. Ensure that the other nodes in the CXFS cluster all have a correct `/etc/failover2.conf` file.

   Note: If there is any doubt that the `/etc/failover2.conf` is correct on the other nodes, ensure there is no I/O from those nodes while performing this procedure (which could move the paths). After completing the procedure, you must fix the files on the other nodes so that paths have the correct affinity

2. Display the available paths between the Windows client and each LUN by running the following command on the Windows client:

   ```
   C:\> xvm show -v phys | find "affinity"
   ```

   For example, paring down the output to just that for LUN 49 (49):

```
C:\> xvm show -v phys  | find "affinity"
21000024FF2468EF-200C00A0B813DF30-49 <dev 3811> affinity=0
21000024FF2468EF-200D00A0B813DF30-49 <dev 2736> affinity=0
21000024FF2468EE-200C00A0B813DF30-49 <dev 1321> affinity=0
21000024FF2468EE-200D00A0B813DF30-49 <dev 246> affinity=0 <current path>
```

   The output has the following format:

   clientHBAcontroller-RAIDcontroller-LUNnumber

   Therefore, the above shows the following (highlighting the differences):

   - The local HBA controllers for the Windows client are:

     21000024FF2468E**F**
     21000024FF2468E**E**

- The RAID controllers are:

  200**C**00A0B813DF30
  200**D**00A0B813DF30

- The LUN is 49

3. Display the available paths between the CXFS metadata server and each LUN by running the following command on the metadata server:

   MDS# **xvm show -v phys | find "affinity"**

   For example, paring down the output to just that for LUN 49:

```
/dev/disk/by-path/pci-0001:00:02.1-fc-0x200d00a0b813df30-lun-49 <sdjf 8:400> affinity=none <current>
/dev/disk/by-path/pci-0001:00:02.1-fc-0x200c00a0b813df30-lun-49 <sdax 67:16> affinity=none
```

   In this case, the metadata server has just one HBA with only one port:

   - pci-0001:00:02.1 identifies the HBA port that is on the metadata server.
   - 0x200d00a0b813df30 identifies the WWPN of the RAID controller.

     The RAID controllers are 200**C**00A0B813DF30 and 200**D**00A0B813DF30

4. Match up the RAID controllers for the Windows client and the metadata server for each LUN.

   Note: The metadata server output will use lowercase and the prefix 0x, while the Windows client output will not include the prefix and will use uppercase.

   For example, for LUN 49, the metadata server RAID controller output of 0x200d00a0b813df30 matches the Windows RAID controller output of 200**D**00A0B813DF30 seen on the Windows client output.

5. Generate a preliminary failover2.conf file. As as shortcut, you can use the output from the following command:

   C:\> **xvm show -v phys | find "affinity" > failover2.conf**

6. Modify the preliminary failover2.conf file you generated in step5 as follows:

a. Set the `affinity` value for a given RAID controller to one value and use another value for the other controller.

Note: Set `affinity` values that are consistent across the cluster.

---

b. Add the `preferred` tag to the preferred paths.

For example, using `affinity=1` for the "`C`" controller:

```
21000024FF2468EF/200C00A0B813DF30/49 <dev 3811> affinity=1
21000024FF2468EF/200D00A0B813DF30/49 <dev 2736> affinity=2
21000024FF2468EE/200C00A0B813DF30/49 <dev 1321> affinity=1 preferred
21000024FF2468EE/200D00A0B813DF30/49 <dev 246> affinity=2
```

7. Copy the `failover2.conf` file to the **CXFS** folder (`%ProgramFiles%\CXFS\`).
8. Run the following `xvm` commands to read in the new configuration and change to the `preferred` path:

```
C:\> xvm foconfig -init
C:\> xvm foswitch -preferred phys
```

Specific RAID (Nonpersistent XVM Device Names)

This procedure applies to the following specific RAID platforms and does not use persistent device naming:

Note: To use persistent device naming in XVM for any of the following that are Fibre Channel RAID, see "FC RAID (Persistent XVM Device Names using WWPNs)" on page 147.

SGI InfiniteStorage 5500
SGI InfiniteStorage 5000
SGI InfiniteStorage 4600
SGI InfiniteStorage 4500
SGI InfiniteStorage 4100
SGI InfiniteStorage 4000
SGI InfiniteStorage 220
SGI TP9700
SGI TP9500S
SGI TP9500

SGI TP9400
SGI TP9300S
SGI TP9300
SGI S330

Note: You must not install RDAC pseudo/virtual LUNs onto the Windows client.

To configure the `failover2.conf` file for the above RAID, do the following:

1. Ensure that the other nodes in the CXFS cluster all have a correct `/etc/failover2.conf` file.

   Note: If there is any doubt that the `/etc/failover2.conf` is correct on the other nodes, ensure there is no I/O from those nodes while performing this procedure (which could move the paths). After completing the procedure, you must fix the files on the other nodes so that paths have the correct affinity

   ---

2. *(QLogic HBA only)* If you have a QLogic HBA, do the following:

   a. Run the SanSurfer utility and set the persistent binding to bind the target (node and port's WWN) to the target ID. For more information, see "Mapping XVM Volumes to Storage Targets on Windows" on page 163.

      Note: You must enable persistent bindings in the QLogic HBA driver for all targets prior to creating the `failover2.conf` file on the Windows system.

      ---

   b. Reboot the Windows node.

3. On the CXFS metadata server, ensure that each RAID is using the preferred path for the primary controller by running the following command:

   ```
   MDS# smeecli -n {RAIDnames} -c "reset storagearray volumedistribution;"
   ```

4. Download the Windows build of `sg_utils` from the following location:

   http://sg.danny.cz/sg/p/sg3_utils-1.33exe.zip

5. Use the `sg_scan.exe` command to get the list of disk devices, where `PD#` is the disk number. For example:

```
C:\> sg_scan.exe
PD0     [C]     SEAGATE   ST3300657SS       0008  6SJ27FJP0000N147BU83
PD1             SEAGATE   ST3300657SS       0008  6SJ23QGS0000N1477GAT
PD2             SGI       IS400             0770  SV04608276
PD3             SGI       IS400             0770  SV04608276
PD4             SGI       Universal Xport   0770  SV04608276
PD5             SGI       IS400             0770  SV04603232
PD6             SGI       IS400             0770  SV04603232
```

   For example, the above output shows that disks 2, 3, 5, and 6 (`PD2`, `PD3`, `PD5`, and `PD6`) are all associated with an SGI RAID product (`IS400`). This indicates that there are four paths to two physical volumes.

6. Use the Windows **Server Manager** to determine the disk properties. Do the following:

   a. Open the **Disk Management** menu:

      Server Manager
      > Storage
      > Disk Management

   b. Do the following for each disk that is associated with SGI RAID:

      i. Click **Properties**. Figure 5-13 shows an example.

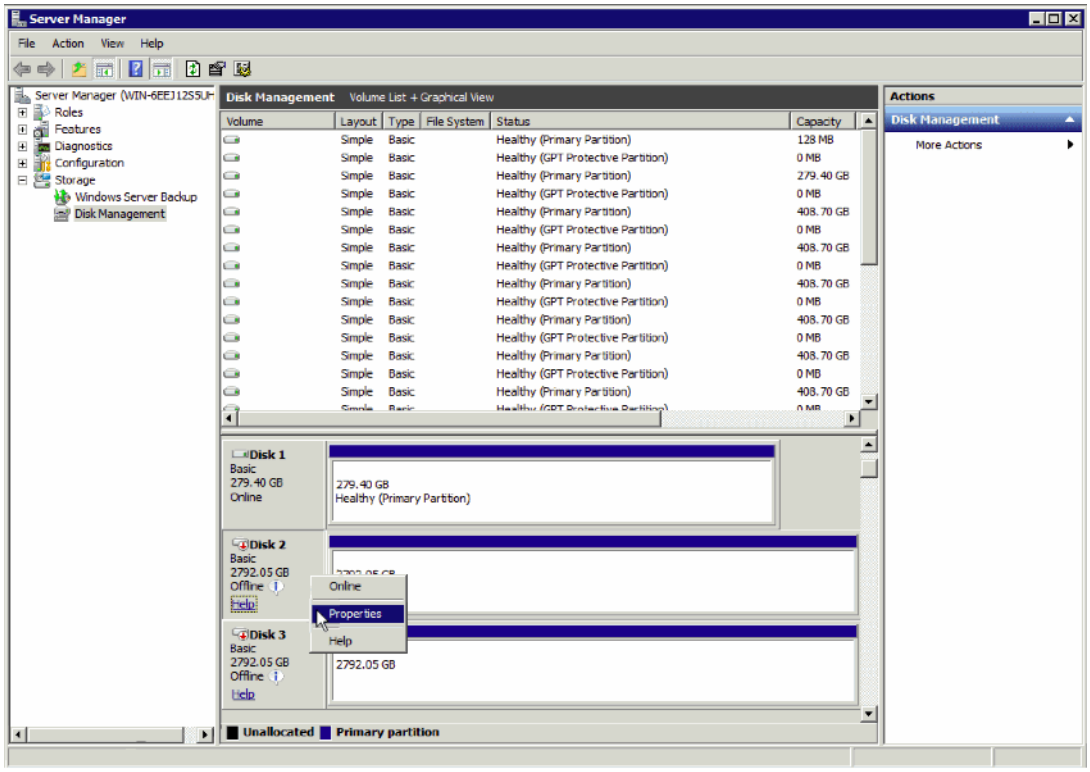

Figure 5-13 Properties Menu

ii. Click the **Details** tab and select the **Device Instance Path** property, as shown in Figure 5-14.

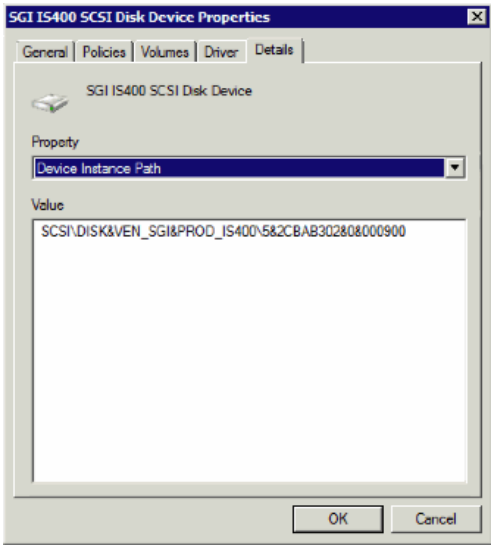

Figure 5-14 Details Tab

For example, Figure 5-14 shows that the value for **Disk 2** (PD2) is:
`SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900`

For the purposes of illustration in steps below, suppose the following additional values:

Disk 3 (PD3):
`SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800`

Disk 5 (PD5): `SCSI\DISK&VEN_SGI&PROD_IS400\000009`
Disk 6 (PD6): `SCSI\DISK&VEN_SGI&PROD_IS400\000008`

7. Determine which paths apply for each XVM physical volume (*physvol*) by using the following command and examining the `available paths` output:

   ```
   C:\> xvm show -v phys
   ```

   For example, highlighting the key lines below `available paths`:

```
C:\> xvm show -v phys
XVM physvol phys/is5500-2_zsf9
========================
```

```
size: 3512172544 blocks  sectorsize: 512 bytes  state:
online,cluster,accessible
uuid: 9fd3a466-06c8-4314-aada-9b801fb22aba
system physvol:  no
physical drive:  SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900 on host cxfs-uv10
available paths:
                 SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900 <dev 106> affinity=0 <current path>
                 SCSI\DISK&VEN_SGI&PROD_IS400\000009 <dev 46> affinity=0
Disk has the following XVM label:
  Clusterid:  0
  Host Name:  myhost
  Disk Name:  is5500-2_zsf9
  Magic:  0x786c6162 (xlab)    Version 2
  Uuid: 9fd3a466-06c8-4314-aada-9b801fb22aba
  last update:    Thu Aug 09 21:33:42 2012
  state: 0xa1<online,cluster,accessible> flags: 0x0<idle>
  secbytes:  512
  label area:  8157 blocks starting at disk block 35 (10 used)
  user area:   3512172544 blocks starting at disk block 8192

Physvol Usage:
Start        Length       Name
--------------------------------------------------
0            3512172544   slice/is5500-2_zsf9s0

XVM physvol phys/is5500-2_zsf10
=========================
size: 3512172544 blocks  sectorsize: 512 bytes  state:
online,cluster,accessible
uuid: 0255793a-b8a4-448a-9ce3-aeb2677a8e37
system physvol:  no
physical drive:  SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800 on host cxfs-uv10
available paths:
                 SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800 <dev 41> affinity=0 <current path>
                 SCSI\DISK&VEN_SGI&PROD_IS400\000008 <dev 66> affinity=0
Disk has the following XVM label:
  Clusterid:  0
  Host Name:  myhost
  Disk Name:  is5500-2_zsf10
  Magic:  0x786c6162 (xlab)    Version 2
  Uuid: 0255793a-b8a4-448a-9ce3-aeb2677a8e37
```

```
  last update:    Thu Aug 09 21:33:42 2012
  state: 0xa1<online,cluster,accessible> flags: 0x0<idle>
  secbytes:  512
  label area:  8157 blocks starting at disk block 35 (10 used)
  user area:   3512172544 blocks starting at disk block 8192

Physvol Usage:
Start        Length       Name
-------------------------------------------------
0            3512172544   slice/is5500-2_zsf10s0
```

The above output shows that the paths apply as follows:

- For physvol `is5500-2_zsf9`:

  ```
  SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900
  SCSI\DISK&VEN_SGI&PROD_IS400\000009
  ```

- For physvol `is5500-2_zsf10`:

  ```
  SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800
  SCSI\DISK&VEN_SGI&PROD_IS400\000008
  ```

Note: By default, XVM assigns a value of `affinity=0` to all paths. In the steps below, you will reassign the values so that paths are grouped appropriately. The path labeled as the `<current path>` is not necessarily the preferred path. For more information, see the sections about `affinity` and the `/etc/failover2.conf` file in *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

---

8. Compare the output from steps 6 and 7 to determine which paths correspond to which disks.

   For example, the information from the preceding steps shows the following:

   - Physvol `is5500-2_zsf9` uses the paths to disks 2 and 5:

     ```
     SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900
     SCSI\DISK&VEN_SGI&PROD_IS400\000009
     ```

   - Physvol `is5500-2_zsf10` uses the paths to disks 3 and 6:

     ```
     SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800
     SCSI\DISK&VEN_SGI&PROD_IS400\000008
     ```

9. Determine which disk is the primary controller:

   a. Display the available vital product data (VPD) pages and note the number of the volume access control page value for the disk:

      ```
      C:\> sg_inq.exe -p 0 PD#
      ```

      For example, the following shows that the volume access control page value for disk 2 (PD2) is 0xc9:

      ```
      # sg_inq -p 0 PD2
       Only hex output supported. sg_vpd decodes more pages.
      VPD INQUIRY, page code=0x00:
         [PQual=0  Peripheral device type: disk]
         Supported VPD pages:
           0x0    Supported VPD pages
           0x80    Unit serial number
           0x83    Device identification
           0x85    Management network addresses
           0x86    Extended INQUIRY data
           0x87    Mode page policy
           0xb1    Block device characteristics (sbc3)
           0xc0    vendor: Firmware numbers (seagate); Unit path report (EMC)
           0xc1    vendor: Date code (seagate)
           0xc2    vendor: Jumper settings (seagate); Software version (RDAC)
           0xc3    vendor: Device behavior (seagate)
           0xc4
           0xc8
           0xc9    Volume Access Control (RDAC)
           0xca
           0xd0
           0xe0
      ```

   b. Query the path priority, using the volume access control page value you determined in step 9a:

      ```
      C:\> sg_inq.exe -p volume_access_control_page PD#
      ```

      For example, for disks 2, 3, 5, and 6:

      ```
      C:\> sg_inq.exe -p 0xc9 PD2
      VPD INQUIRY: Volume Access Control (RDAC)
        AVT: Enabled
        Volume Access via: primary controller
      ```

```
  Path priority: 1 (preferred path)
C:\> sg_inq.exe -p 0xc9 PD3
VPD INQUIRY: Volume Access Control (RDAC)
  AVT: Enabled
  Volume Access via: primary controller
  Path priority: 1 (preferred path)
C:\> sg_inq.exe -p 0xc9 PD5
VPD INQUIRY: Volume Access Control (RDAC)
  AVT: Enabled
  Volume Access via: alternate controller
  Path priority: 2 (secondary path)
C:\> sg_inq.exe -p 0xc9 PD6
VPD INQUIRY: Volume Access Control (RDAC)
  AVT: Enabled
  Volume Access via: alternate controller
  Path priority: 2 (secondary path)
```

For example, the preceding steps indicate the following:

- Physvol `is5500-2_zsf9`:
  - Disk 2 (`SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900`) is the primary controller and the preferred path
  - Disk 5 (`SCSI\DISK&VEN_SGI&PROD_IS400\000009`) is the alternate controller
- Physvol `is5500-2_zsf10`:
  - Disk 3 (`SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800`) is the primary controller and the preferred path
  - Disk 6 (`SCSI\DISK&VEN_SGI&PROD_IS400\000008`) is the alternate controller

10. Generate a preliminary `failover2.conf` file. As as shortcut, you can use the output from the following command:

    ```
    C:\> xvm show -v phys | find "affinity" > failover2.conf
    ```

11. Modify the `failover2.conf` file you generated in step 10 as follows according to the results of step 9:

    a. Set `affinity=1` for the primary controllers and set `affinity=2` for the secondary controllers.

Note: Set `affinity` values that are consistent across the cluster.

b. Add the `preferred` tag to the preferred paths.

For example:

```
SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000900 <dev 106> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_IS400\000009 <dev 46> affinity=2
SCSI\DISK&VEN_SGI&PROD_IS400\5&2CBAB302&0&000800  <dev 41> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_IS400\000008 <dev 66> affinity=2
```

12. Copy the `failover2.conf` file to the **CXFS** folder (`%ProgramFiles%\CXFS\`).
13. Run the following `xvm` commands to read in the new configuration and change to the `preferred` path:

```
C:\> xvm foconfig -init
C:\> xvm foswitch -preferred phys
```

For more information, see:

- "XVM Failover and CXFS" on page 8
- The comments in the `failover2.conf` file
- *CXFS 7 Administrator Guide for SGI InfiniteStorage*
- *XVM Volume Manager Administrator Guide*

Other RAID (Nonpersistent XVM Device Names)

Note: XVM device names will persist for QLogic HBAs only. You can also use this method for ATTO and LSI HBAs, but you may have to modify the `failover2.conf` file after rebooting.

Do the following for QLogic HBAs, ATTO HBAs, and LSI HBAs:

1. Run the SanSurfer utility and set the persistent binding to bind the target (node and port's WWN) to the target ID. For more information, see "Mapping XVM Volumes to Storage Targets on Windows" on page 163.

Note: For the `failover2.conf` file to work properly, persistent bindings must be enabled in the QLogic HBA driver.

2. Find the WWN of the corresponding port and node (controller) on the storage array. As a result, a target ID corresponds to a controller and a port on the controller.

   Note: You must make sure that the `failover2.conf` setting is consistent across the cluster.

   In the persistent binding, there are normally the following fields:

   - Type
   - Target's node WWN (the controller's WWN)
   - Target's port WWN (the port on the controller)
   - A configurable target ID

3. Note the controller and port to which the target ID corresponds.
4. Reboot the Windows node.

For example, assume there are two controllers in a storage array. Controller A has a WWN of `200400a0b82925e2`; it has two ports connecting to the host or the fabric. Port 1 has a WWN of `201400A0B82925E2`, port 2 has a WWN of `202400A0B82925E2`. Controller B has a WWN of `200500a0b82925e2`; it also has two ports with WWNs of `201500A0B82925E2` and `202500A0B82925E2`, respectively. There are therefore four paths to LUN 0.

The metadata server in this cluster would have entries like the following in its `failover2.conf` file (where information within angle brackets is an embedded comment):

```
/dev/xscsi/pci08.03.1/node200500a0b82925e2/port2/lun0/disc affinity=2
/dev/xscsi/pci08.03.1/node200500a0b82925e2/port1/lun0/disc affinity=2
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port2/lun0/disc affinity=1
/dev/xscsi/pci08.03.1/node200400a0b82925e2/port1/lun0/disc affinity=1 preferred <current path>
```

In this configuration, controller A (`node200400a0b82925e2`) has an affinity of `1`, controller B has an affinity of `2`. Controller A's port 1 is the preferred path.

To create the corresponding `failover2.conf` file on the Windows node, you must first define the persistent-binding targets. Use SANSurfer (for Qlogic HBA) or LSIUtil (for LSI HBA) to define four possible targets:

```
Binding type     World Wide Node Name   World Wide port Name   Target ID
WWN             200500a0b82925e2       202500A0B82925E2        0
WWN             200500a0b82925e2       201500A0B82925E2        1
WWN             200400a0b82925e2       202400A0B82925E2        2
WWN             200400a0b82925e2       201400A0B82925E2        3
```

As a result, target `0` corresponds to the first path on the metadata server. Targets `1`, `2`, and `3` correspond to the 2nd, 3rd, and 4th path, respectively. To be consistent, target `2` or `3` (on controller A) should be the preferred path on Windows.

Then you would run the following command:

```
C:\> xvm show -v phys | find "affinity" > failover2.conf
```

Assuming that there are two HBA ports on the Windows node, you would end up with eight paths for the two HBA ports. The `failover2.conf` file would contain something like the examples shown in the following sections (the format varies by the Windows OS version):

- "Windows XP SP2 and Windows Server 2003 R2 SP1 `failover2.conf` Example " on page 161
- "Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 `failover2.conf` Example" on page 162

Windows XP SP2 and Windows Server 2003 R2 SP1 `failover2.conf` Example

Windows XP SP 2 and Windows Server 2003 R2 SP1 `failover2.conf` example:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&020 <dev 301> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&010 <dev 281> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000 <dev 261> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&030 <dev 236> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&020 <dev 216> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&010 <dev 196> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000 <dev 176> affinity=0
#
#  Where
#  SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=0
#                                         ^^^^^^^   ^^^
#                                            |      |||-- Lun = 0
#                                            |      ||--- Target = 1 (1-2 hex digits)
#                                            |      |---- Bus ID = 0
#                                            |------------ Host HBA port ID = 67032E4
```

You would set the proper `affinity` values and add the `preferred` tag to target `2` or `3`:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&030 <dev 321> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&020 <dev 301> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&010 <dev 281> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000 <dev 261> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&030 <dev 236> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&020 <dev 216> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&010 <dev 196> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000 <dev 176> affinity=2
```

In this setting, the access to LUN 0 from one HBA (with its ID of `67032E4`) goes to controller A, port 1. From another HBA (with ID of `1F095A8E`), it goes to controller A, port 2. Controller A (to which targets `2` and `3` belong) has an affinity of `1`; controller B has an affinity of `2`.

Windows Server 2003 R2 SP2, Windows Vista, Windows Server 2008, and Windows 7 `failover2.conf` Example

Windows Server 2003 R2 SP2, Windows Vista, and Windows Server 2008 `failover2` example:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000300 <dev 321> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000200 <dev 301> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000100 <dev 281> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000000 <dev 261> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000300 <dev 236> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000200 <dev 216> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000100 <dev 196> affinity=0
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000000 <dev 176> affinity=0
#
#  Where
#  SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&00300 <dev 321> affinity=0
#                                          ^^^^^^^   ^^^^^
#                                             |      || |- Lun = 0   (2 hex digits)
#                                             |      ||--- Target = 3 (2 hex digits)
#                                             |      |---- Bus ID = 0
#                                             |------------ Host HBA port ID = 67032E4
```

You would set the proper `affinity` values and add the `preferred` tag to target 2 or 3:

```
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000300 <dev 321> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000200 <dev 301> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000100 <dev 281> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&67032E4&0&000000 <dev 261> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000300 <dev 236> affinity=1
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000200 <dev 216> affinity=1 preferred
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000100 <dev 196> affinity=2
SCSI\DISK&VEN_SGI&PROD_TP9700&REV_0619\5&1F095A8E&0&000000 <dev 176> affinity=2
```

In this setting, the access to LUN 0 from one HBA (with its ID of `67032E4`) goes to controller A, port 1. From another HBA (with ID of `1F095A8E`), it goes to controller A, port 2. Controller A (to which targets 2 and 3 belong) has an affinity of 1; controller B has an affinity of 2.

Converting an Existing `failover2.conf` File for Windows with FC RAID with RAID

To convert an existing `failover2.conf` for Fibre Channel RAID to use persistent pathnames, do the following:

1. Ensure that the current configuration is loaded into the kernel and uses the preferred physvols:

   ```
   C:\> xvm foconfig -init
   C:\> xvm foswitch -preferred phys
   ```

2. Copy the existing `failover2.conf` to a new location as a safety measure in case you want to restore it.

3. Create a new `failover2.conf` file:

   ```
   C:\> xvm show -v phys | find "affinity" > failover2.conf
   ```

Configuring I/O Fencing for Windows (FC Only)

I/O fencing is required on Windows nodes in order to protect data integrity of the filesystems in the cluster. The CXFS client software automatically detects the worldwide port names (WWPNs) of any supported FC HBAs for Windows nodes that are connected to a switch that is configured in the cluster database. These HBAs are available for fencing.

Mapping XVM Volumes to Storage Targets on Windows

You must configure the HBA on each node to use persistent bindings for all ports used for CXFS filesystems. The method for configuration varies depending on your HBA vendor. For more information, see the following:

- Information about binding target devices is in the QLogic SANsurfer help. You must select a port number and then select **Bind** and the appropriate **Target ID** for each disk. For example, see Figure 5-15.
- Information about persistent bindings is in the LSI Logic MPT Configuration Utility (`LSIUtil.exe`). `LSIUtil` is a command line tool. It has a submenu for displaying and changing persistent mapping. Do the following:

  1. Choose the HBA port.
  2. Select **e** to enable expert mode.

3. Select **15** to manipulate persistent binding.
4. Choose one of the following:
   - **2** to automatically add persistent mappings for all targets
   - **3** to automatically add persistent mappings for some targets
   - **6** to manually add persistent mappings

Note: You should disable any failover functionality provided by the HBA.

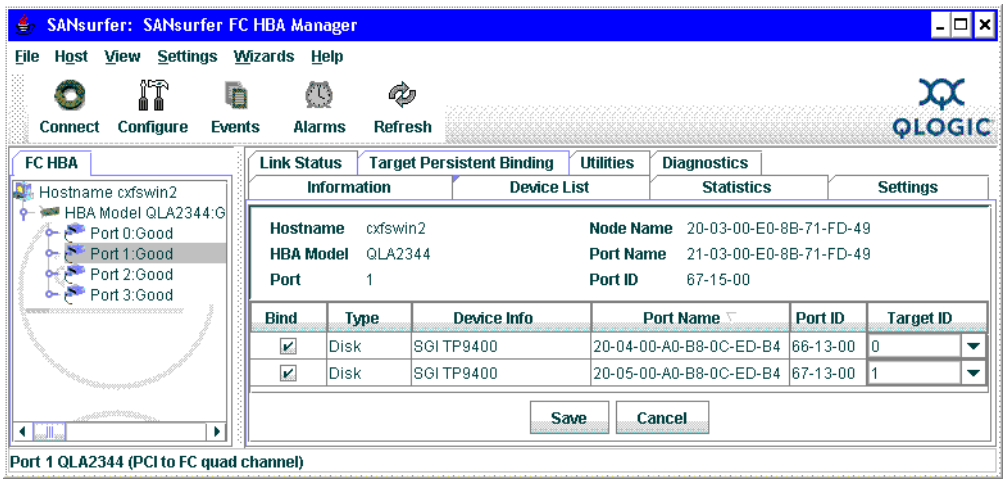

Figure 5-15 QLogic SANsurfer (Copyright QLogic® Corporation, all rights reserved)

Start/Stop the `cxfs_client` Service for Windows

By default, the `cxfs_client` service is automatically started when a Windows node is rebooted, and it starts the CXFS filesystem drivers. SGI recommends that you do not change this default for a production system.

However, if for testing purposes you want to manually start/stop the service, do the following:

1. Select the following:

   Start
   > Control Panel
   > Administrative Tools
   > Services

2. Change **CXFS Client** to manual.

You can then manually start/stop CXFS by using the same selection sequence.

Maintenance for Windows

This section contains the following:

Modifying CXFS Folder Permissions on Windows

Permissions for folders do not appear directly in the **Security** tab under **Permissions for *username*** because permissions are inheritable by default in NTFS but not in CXFS. The following sections discuss how to modify CXFS folder permissions:

Modifying Permissions: Windows 8 and Later

For Windows 8, 8.1, Windows Server 2012, and Windows Server 2012 R2, do the following to modify CXFS folder permissions from a Windows client:

1. Right-click on a folder in **Explorer** and select **Properties**.
2. Navigate to the **Security** tab.
3. Click the **Advanced** button.
4. In the **Permission entries** box of the **Advanced Security Settings for *foldername*** dialog that appears, select the **Principal** for which permissions should be changed. Click the **Edit** button.
5. In the **Permission Entry for *foldername*** dialog that appears, check or uncheck the permissions as desired.
6. Click **OK** in the **Permission Entry for *foldername*** dialog, the **Advanced Security Settings for *foldername*** dialog, and the Properties dialog.

Modifying Permissions: Windows 7, Windows Vista, Windows 2008, and Windows 2008 R2

For Windows 7, Windows Vista, Windows 2008, and Windows 2008 R2 do the following to modify CXFS folder permissions from a Windows client:

1. Right-click on a folder in **Explorer** and select **Properties**.
2. Navigate to the **Security** tab.
3. Click the **Advanced** button.
4. In the **Permission entries** box of the **Advanced Security Settings for *foldername*** dialog that appears, select the **Name** for which permissions should be changed. Click the following:

   - Windows 7 or Windows 2008 R2:

     Change
     > Edit

- Windows Vista and Windows 2008:

  Edit
  > Edit

5. In the **Permission Entry for *foldername*** dialog that appears, check or uncheck the permissions as desired.
6. Click **OK** in the **Permission Entry for *foldername*** dialog, the **Advanced Security Settings for *foldername*** dialog, and the Properties dialog.

Modifying Permissions: Windows XP, Windows Server 2003, and Windows Server 2003 R2

For Windows XP, Windows Server 2003, and Windows Server 2003 R2, do the following to modify CXFS folder permissions from a Windows client:

1. Right-click on a folder in **Explorer** and select **Properties**.
2. Navigate to the **Security** tab.
3. Click the **Advanced** button.
4. In the **Permission entries** box of the **Advanced Security Settings for *foldername*** dialog that appears, select the **Name** for which permissions should be changed. Click the **Edit** button.
5. In the **Permission Entry for *foldername*** dialog that appears, check or uncheck the permissions as desired.
6. Click **OK** in the **Permission Entry for *foldername*** dialog, the **Advanced Security Settings for *foldername*** dialog, and the Properties dialog.

Modifying the CXFS Software for Windows

To change the location of the software and other configuration settings that were requested in "Client Software Installation for Windows" on page 136, perform the following steps:

1. Select the following:

   - Windows XP and Windows 2003:

     Start
     > Control Panel
     > Add or Remove Programs
     > CXFS
     > Add/Remove
     > Modify

   - Windows Vista, Windows Server 2008, and Windows 7:

     Start
     > Control Panel
     > Programs and Features
     > CXFS
     > Change

   Figure 5-16 shows the screen that lets you modify the software.

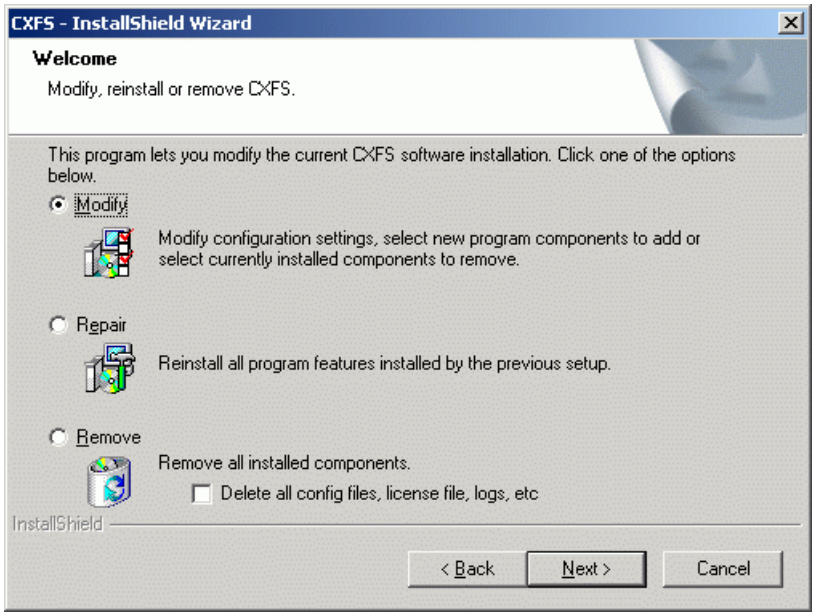

Figure 5-16 Modify CXFS for Windows

2. Make the necessary configuration changes.

   You can display the list of possible command line arguments supported by the `cxfs_client` service by running the service from a command line as follows:

   ```
   C:\> %SystemRoot%\system32\cxfs_client.exe -h
   ```

3. Reboot the system to apply the changes.

Updating the CXFS Software for Windows

To upgrade the CXFS for Windows software, perform the following steps:

1. Obtain the CXFS update software according to the directions in the *CXFS 7 Administrator Guide for SGI InfiniteStorage* and the *SGI InfiniteStorage Software Platform* release notes.

2. Transfer the client software (which was downloaded onto a server-capable administration node during its installation procedure) using `ftp`, `rcp`, or `scp`. The location of the Windows installation program will be as follows:

`/usr/cluster/client-dist/`*CXFS_VERSION*`/windows/all/noarch/setup.exe`

3. Double-click the **setup.exe** installation program to execute it.

4. A welcome screen will appear that displays the version you are upgrading from and the version you are upgrading to. Figure 5-17 shows an example of the screen that appears when you are upgrading the software (the actual versions displayed by your system will vary based upon the release that is currently installed and the release that will be installed. All the configuration options are available to update as discussed in "Client Software Installation for Windows" on page 136.

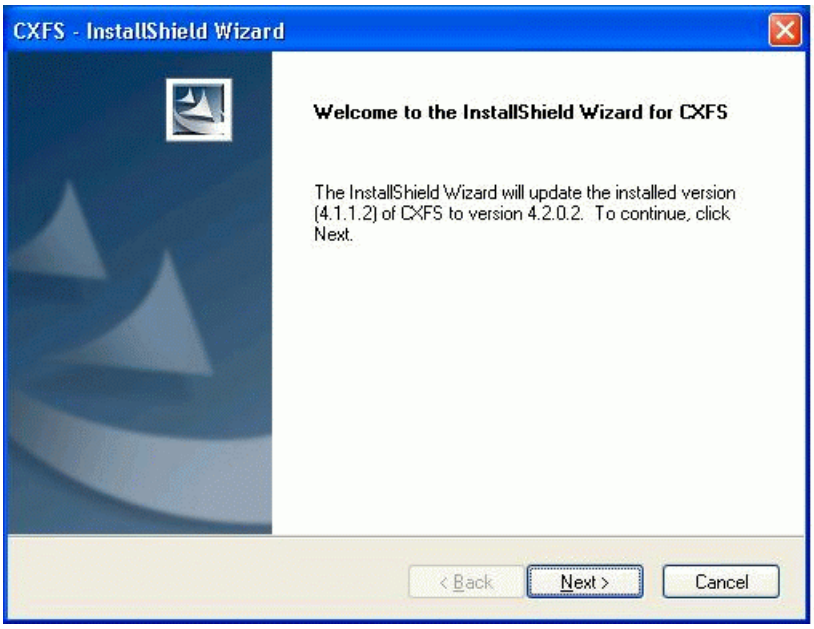

Figure 5-17 Upgrading the Windows Software

5. Reboot the system to apply the changes.

Removing the CXFS Software for Windows

To remove the CXFS for Windows software, do the following:

1. Ensure that no applications on this node are accessing files on a CXFS filesystem.
2. Select the following sequence to remove all installed files and registry entries:
   - Windows XP and Windows 2003:

     Start
     > Control Panel
     > Add or Remove Programs
     > CXFS
     > Add/Remove
     > Remove

   - Windows Vista, Windows Server 2008, and Windows 7:

     Start
     > Control Panel
     > Programs and Features
     > CXFS
     > Remove

   Figure 5-16 on page 169 shows the screen that lets you remove the software.

   Note: By default, the `passwd`, `group`, and `log` files will not be removed. To remove these other files, check the following box:

   Delete all config files, license file, logs, etc

   Then click **Next**.

   ---

3. Reboot the system to apply the changes.

Downgrading the CXFS Software for Windows

To downgrade the CXFS software, do the following:

1. Back up the configuration file.

Note: The removal process may remove the configuration file. You should back up the configuration file before removing the CXFS software so that you can easily restore it after installing the downgrade.

2. Follow the instructions to remove the software in "Removing the CXFS Software for Windows" on page 171.
3. Install the older version of the software as directed in "Client Software Installation for Windows" on page 136.

GRIO on Windows

CXFS supports guaranteed-rate I/O (GRIO) version 2 on the Windows platform if GRIO is enabled on the server-capable administration node.

Figure 5-18 shows an example of the **CXFS Info** display for GRIO.

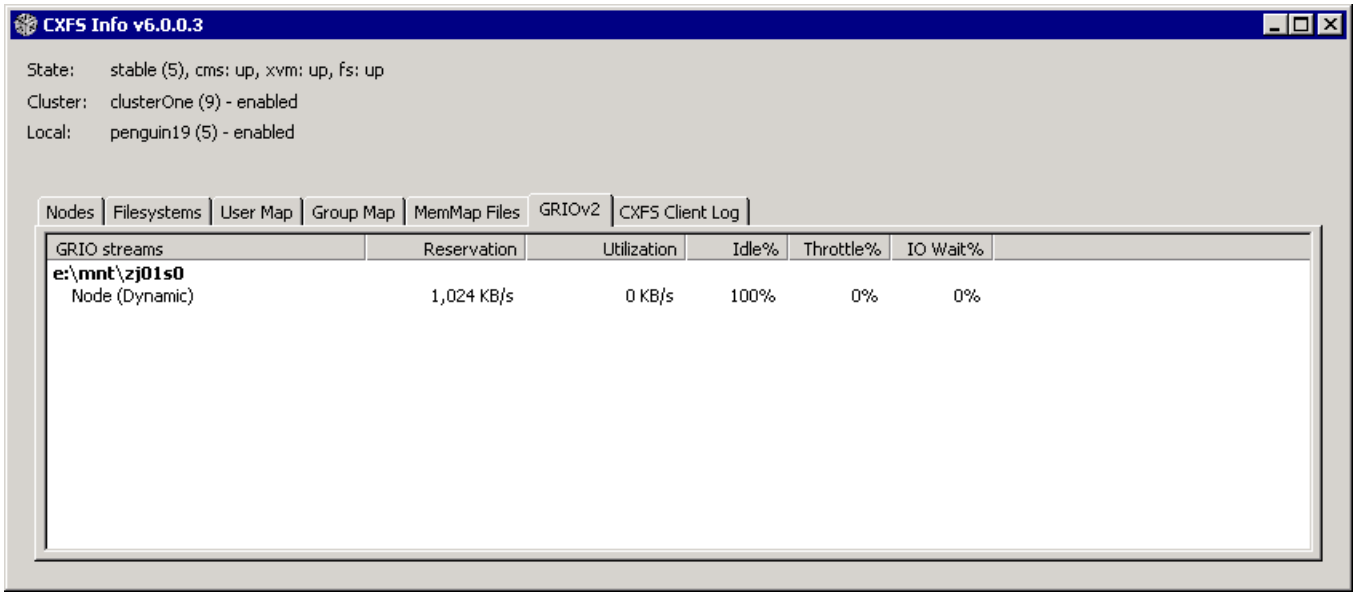

Figure 5-18 CXFS Info Display for GRIO for Windows

A Windows node can mount a GRIO-managed filesystem and supports application- and node-level reservations. A Windows node will interoperate with the dynamic bandwidth allocator for all I/O outside of any reservation.

For more information, see "Guaranteed-Rate I/O (GRIO) and CXFS" on page 7 and the *Guaranteed-Rate I/O Version 2 for Linux Guide*.

System-Tunable Parameters for Windows

SGI recommends that you use the same settings for kernel system tunable parameters on all applicable nodes in the cluster.

Caution: You should only change system tunable parameters if you are fully aware of their consequences or if directed to do so by SGI Support.

This section discusses the following topics:

Note: These system tunables are removed when the software is removed. They may need to be reset when downgrading the CXFS for Windows software.

Overview of Registry Modification

In order to configure system tuning settings, you must modify the registry using the `regedit.exe` program to add registry settings. Do the following:

1. Back up the registry before making any changes.
2. Open the `regedit.exe` program.
3. Add or modify the registry setting with the desired value.
4. Reboot the system to apply the changes.

Caution: Only the entries documented here may be changed to modify the behavior of CXFS. All other registry entries for CXFS must not be modified or else the software may no longer function.

Tuning the Verbosity of CXFS Messages in the System Event Log for Windows

You can specify the level of verbosity for CXFS messages that are logged to the System Event log by changing the following `DWORD` in the registry to a value that specifies the desired level of verbosity:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\LogVerbosity
```

By default, it is set to `4`, which is fairly verbose. The higher the number, the more messages that are logged. You can reset the value to one of the following, as appropriate for your site:

Value	Events Logged
`0`	None (disables the logging of all events from the `cxfs_client` service)
`1`	Panic events only
`2`	Alert and panic events
`3`	Warning, alert, and panic events

4	Notice, warning, alert, and panic events (default)
5	Informational, notice, warning, alert, and panic events
6	Debug, informational, notice, warning, alert, and panic events events

Note: If you enter a value that is not in the range 0 through 6, it will be rejected and the cxfs_client service will then use the default value of 4 instead.

Default Umask for Windows

The default umask that is set up during installation can be configured to a value not supported by the installer. Edit the following DWORD in the registry to an appropriate umask:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\DefaultUmask
```

Note: This value specifies the umask in hexadecimal or decimal, not its normal octal representation used on UNIX platforms.

For more information on the umask, see "Inheritance and Default ACLs for Windows" on page 131.

Maximum DMA Size for Windows

By default, CXFS for Windows breaks down large direct I/O requests into requests no larger than 16 MB. You can change the size of these requests by adding the following DWORD to the registry with a value that specifies the maximum I/O request size in bytes:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\MaxDMASize
```

Memory-Mapping Coherency for Windows

By default, a CXFS Windows node enforces memory-mapping coherency by preventing other clients and the CXFS metadata server access to the file while it is mapped. This can cause problems for some applications that do not expect this behavior.

Microsoft Office applications and `Notepad.exe` use memory-mapped I/O to read and write files, but use byte-range locks to prevent two people from accessing the same file at the same time. The CXFS behavior causes the second Office application to hang until the file is closed by the first application, without displaying a dialog that the file is in use.

Backup applications that search the filesystem for modified files will stall when they attempt to back up a file that has been memory-mapped on a CXFS Windows node.

You can change the following `DWORD` in the registry from `0` to `1` to avoid these problems:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\DisableMemMapCoherency
```

Caution: However, if you set `DisableMemMapCoherency` to `1`, CXFS can no longer ensure data coherency if two applications memory-map the same file at the same time on different nodes in the cluster. Use this option with extreme caution with multiple clients concurrently accessing the same files.

Also see "DMF and Memory-Mapped Files on Windows" on page 118.

DNLC Size for Windows

The Directory Name Lookup Cache (DNLC) in a CXFS Windows node allows repetitive lookups to be performed without going to the metadata server for each component in a file path. This can provide a significant performance boost for applications that perform several opens in a deep directory structure.

The `DnlcSize` parameter is set to `4096` by default. You can change it to a value from `0` (which disables the DNLC) to `100000` (values outside this range will be reset to `4096`). Edit the following `DWORD` registry value:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\DnlcSize
```

Note: Increasing the DNLC size can have a significant memory impact on the Windows node and the metadata server because they maintain data structures for every actively opened file on the CXFS clients. You should monitor the memory usage on these nodes before and after changing this parameter because placing nodes under memory pressure is counter-productive to increasing the DNLC size.

Mandatory Locks for Windows

By default, byte-range locks across the cluster are advisory locks, which do not prevent a rogue application from reading and writing to locked regions of a file.

Note: Windows filesystems (NTFS and FAT) implement a mandatory locking system that prevents applications from reading and writing to locked regions of a file. Mandatory locks are enabled within a Windows node.

To enable mandatory byte-range locks across the cluster, set the following `DWORD` value to `1`:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\ForceMandatoryLocks
```

Note: Setting this parameter will adversely affect performance of applications using these locks.

User Identification Map Updates for Windows

User identification maps are updated automatically by the following triggers:

- An unmapped user logs into the system
- The `passwd` and/or `group` file is modified when the primary mapping method is **files**
- An LDAP database change is detected when the primary mapping method is **ldap_activedir** or **ldap_generic**

The most common trigger in a typical environment is when an unmapped user logs into the system; the other two triggers are generally static in nature.

Updating the map can be a resource-intensive operation in a domain environment. Therefore, by default, an update is triggered only when an unmapped user logs in and not more often than every 5 minutes.

To configure the minimum update interval, add the following `DWORD` to the registry with a value that is the minimum time between updates in minutes (minimum allowed time is 1 minute):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS_Client\Parameters\MinMapGenTime
```

I/O Size Issues Within the QLogic HBAs

The maximum size of I/O issued by the QLogic HBA defaults to only 256 KB. Many applications are capable of generating much larger requests, so you may want to increase this I/O size to the HBA's maximum of 1 MB.

To increase the size of the I/O, change the setting of the `Device` parameter to a value from `16` through `255` (`0x10` hexadecimal to `0xFF`). A value of 255 (`0xFF`) enables the maximum 1–MB transfer size. Setting a value higher than 255 results in 64–KB transfers. The default value is `33` (`0x21`). Edit the following `DWORD` registry value:

`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ql2`*xxx*`\Parameters\Device`

Command Tag Queueing (CTQ)

Command Tag Queueing (CTQ) is used by HBAs to manage the number of outstanding requests each adapter port has to each target. Adjusting this value (up or down) can improve the performance of applications, depending on the number of clients in the cluster and the number of I/O requests they require to meet the required quality of service.

You should only modify this setting for HBA ports that are to be used by CXFS. Do not modify ports used for local storage.

While it is possible to change this value with the volume mounted, I/O will halt momentarily and there may be problems if the node is under a heavy load.

Note: The Windows HBA may not recognize the CTQ setting placed on the disk by Linux nodes.

To configure the CTQ, use the management tool provided by the HBA. You may also be able to set the execution throttle in the HBA BIOS during boot-up by pressing a key combination when you see the HBA's BIOS message. You should use an execution throttle value (that is, how many commands will be queued by the HBA) in the range 1 through 256. For more information, see the HBA card and driver documentation.

Note: Unlike CTQ, you cannot have separate depths per LUN. Execution throttle limits the number of simultaneous requests for **all** targets in the specified port.

Heartbeat Period

To change the heartbeat period on the Windows node, set the heartbeat period to the desired value (in seconds). You should only change this value at the recommendation of SGI support. The same value must be used on all nodes in the cluster. Edit the following DWORD registry value:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS\Parameters\HeartBeatPeriod
```

For more information, see the section about `mtcp_hb_period` in the *CXFS 7 Administrator Guide for SGI InfiniteStorage.*

Delay Automatic Start of the CXFS Client (Windows Vista and Later)

For Windows Vista and later, you can delay the automatic start of the CXFS client. To enable the delay (assuming that the `Start` value in the same key is already set to `2`, which is the default), add the following `DWORD` to the registry with a value of `1`:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CXFS_Client\DelayedAutoStart
```

To disable the delay, set the value to `0` (the default).

To enable the delay using the UI, do the following:

1. Run `services.msc` or open the following:

   Start
   > Control Panel
   > System

   Maintenance
   > Administrative Tools
   > Services

2. Double-click **CXFS Client**
3. In the **Startup type** drop-down, select **Automatic (Delayed Start)**
4. Click **OK** or **Apply**.

To adjust the time interval before CXFS starts (and any other services with a start type of `DelayedAutoStart`), add the following `DWORD` to the registry using a value set to the desired number of seconds (the default is `120`):

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\AutoStartDelay
```

Troubleshooting for Windows

This section discusses the following:

Also see:

- The Windows `cxfsdump` documentation located at `%ProgramFiles%\CXFS\cxfsdump.html`

- Chapter 7, "General Troubleshooting" on page 213

Verification that the CXFS Software is Running Correctly for Windows

To verify that the `cxfs_client` service has started, select the following:

Start
> Control Panel
> Administrative Tools
> Services

Inability to Mount Filesystems on Windows

If **CXFS Info** reports that `cms` is up but XVM or the filesystem is in another state, then one or more mounts is still in the process of mounting or has failed to mount.

The CXFS node might not mount filesystems for many reasons, including the following:

- The metadata server is unable to mount the filesystem. In this case, no clients will be able to mount the filesystem.
- The metadata server is processing a recovery or relocation that is not progressing. In this case, clients cannot mount the filesystem until that state is cleared. Clearing the state may require you to take action on one or more of the other nodes in the cluster.
- The node may not be able to see all the LUNs. This is usually caused by misconfiguration of the HBA or the SAN fabric:
  - Check that the ports on the Fibre Channel switch connected to the HBA are active. Physically look at the switch to confirm the light next to the port is green, or remotely check by using the `switchShow` command.
  - Check that the HBA configuration is correct.
  - Check that the HBA can see all the LUNs for the filesystems it is mounting.
  - Check that the operating system kernel can see all the LUN devices. For example:

Start
> Control Panel
> Administrative Tools
> ComputerManagement
> Device Manager
> View
> Devices by connection

– Use `debugview` to monitor the `cxfs_client` service when it probes the disk devices. You should see it successfully probe each of the LUN devices.

Note: For Windows Vista and later: By default, debug messages are turned off. To view debug messages using `debugview`, you must enable them in the registry. Add the following `DWORD` to the registry with a value of `0xF`:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Debug Print Filter\DEFAULT
```

– If the RAID device has more than one LUN mapped to different controllers, ensure the node has a Fibre Channel path to all relevant controllers.

- The `cxfs_client` service may not be running. To verify that it is running, open the **Task Manager** by pressing the `Ctrl+Shift+Esc`, or right-mouse click an empty area of the taskbar and select **Task Manager** from the popup menu. In the **Processes** tab, search for `cxfs_client.exe` in the **Image Name** column. You can sort the processes by name by clicking the heading of the column.
- The filesystem may have an unsupported mount option. Check the `cxfs_client.log` for mount option errors or any other errors that are reported when attempting to mount the filesystem.
- The cluster membership (`cms`), XVM, or the filesystems may not be up on the node. Use **CXFS Info** to determine the current state of `cms`, XVM, and the filesystems. Do the following:
  – If `cms` is not up, check the following:
    - Is the node is configured on the server-capable administration node with the correct hostname or IP address?
    - Has the node been added to the cluster and enabled? See "Verifying the Cluster Status" on page 205.
  – If XVM is not up, check that the HBA is active and can see the LUNs.

  - If the filesystem is not up, check that one or more filesystems are configured to be mounted on this node and check the **CXFS Client Log** tab in **CXFS Info** for mount errors. They will be highlighted.

- The LUN may be too large. Windows XP does not support LUNs greater than 2 TB in size. Filesystem corruption will occur if you attempt to write to the LUN above the 2-TB boundary. On a Windows XP node, CXFS will not allow a filesystem to be mounted if any part of it resides on a LUN that is greater than 2-TB in size.

Also, check the **CXFS Client Log** tab in **CXFS Info** for mount errors.

Access-Denied Error when Accessing Filesystem on Windows

If an application reports an access-denied error, do the following:

- Check the list of users and groups that **CXFS Info** has mapped to a UNIX UID and GID. If the current user is not listed as one of those users, check that the user mapping method that was selected is configured correctly, that there is an LDAP server running (if you are using LDAP), and that the user is correctly configured.
- Increase the verbosity of output from the `cxfs_client` service so that it shows each user as it is parsed and mapped.
- Use Process Monitor to monitor the application and verify that there is no file that has been created below a mount point under the CXFS drive letter. An error may be caused by attempting to create a file below the drive letter but above the mount point. For more information, see:

  http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx

Application Works with NTFS but not CXFS for Windows

The Windows filesystem APIs are far more extensive than the UNIX POSIX APIs and there are some limitations in mapping the native APIs to POSIX APIs (see "Functional Limitations and Considerations for Windows" on page 111). Sometimes these limitations may affect applications, other times the applications that have only ever been tested on NTFS make assumptions about the underlying filesystem without querying the filesystem first.

If an application does not behave as expected, and retrying the same actions on an NTFS filesystem causes it to behave as was expected, then third-party tools like Process Monitor can be used to capture a log of the application when using both

NTFS and CXFS. Look for differences in the output and try to determine the action and/or result that is different. Using the same filenames in both places will make this easier. For more information about Process Monitor, see:

http://technet.microsoft.com/en-us/sysinternals/bb896642.aspx

---

Note: There are some problems that may not be visible in a Process Monitor log. For example, some older applications use only a 32-bit number when computing filesystem or file size. Such applications may report out of disk space errors when trying to save a file to a large (greater than 1 TB) filesystem.

Delayed-Write Error Dialog is Generated by the Windows Kernel

A delayed-write error is generated by the Windows kernel when it attempts to write file data that is in the cache and has been written to disk, but the I/O failed. The write call made by the application that wrote the data may have completed successfully some time ago (the application may have even exited by now), so there is no way for the Windows kernel to notify the application that the I/O failed.

This error can occur on a CXFS filesystem if CXFS has lost access to the disk due to the following:

- Loss of membership resulting in the Windows node being fenced and the filesystem being unmounted. Check that the Windows node is still in membership and that there are no unmount messages in the `cxfs_client.log` file.
- Loss of Fibre Channel connection to the Fibre Channel switch or RAID. Check the Fibre Channel connections and use the SanManager tool to verify that the HBA can still see all of the LUNs. Make sure the filesystems are still mounted.
- The metadata server returned an I/O error. Check the system log on the metadata server for any I/O errors on the filesystem and take corrective action on the server if required.

cxfs_client Service Does Not Start on Windows

The following error may be seen when the `cxfs_client` service attempts to start:

```
Error 10038: An operation was attempted on something that is not a socket.
```

Check the **CXFS Client Log** in **CXFS Info** for information on why the CXFS node failed to start.

cxfs_client Service Cannot Map Users other than `Administrator` for Windows

If the `cxfs_client` service cannot map any users other than `Administrator` and there are no LDAP errors in the `cxfs_client` log file (and you are using LDAP), you must change the configuration to allow reading of the attributes.

Do the following:

1. Select the following:

   Start
   > Control Panel
   > Administrative Tools
   > Active Directory Users and Computers

2. Select the following:

   View
   > Advanced Features

3. Right-mouse click the **Users** folder under the domain controller you are using and select the following:

   Properties
   > Security
   > Advanced
   > Add

4. Select `Authenticated Users` from the list and click **OK**.

5. Select `Child Objects Only` from the **Apply onto** drop-down list and check `Read All Properties` from the list of permissions.

6. Click **OK** to complete the operation.

If the above configuration is too broad security-wise, you can enable the individual attributes for each user to be mapped.

Filesystems Are Not Displayed on Windows

If the CXFS drive letter is visible in Windows Explorer but no filesystems are mounted, do the following:

- Run `%ProgramFiles%\CXFS\cxfs_info` to ensure that the filesystems have been configured for this node.
- Verify the filesystems that should be mounted. For more information, see "Mounting Filesystems" on page 202.
- Ensure that the CXFS metadata server is up and that the Windows node is in the cluster membership; see "Verifying the Cluster Status" on page 205.
- Check that the `cxfs_client` service has started. See "Start/Stop the `cxfs_client` Service for Windows" on page 164 and "Verification that the CXFS Software is Running Correctly for Windows" on page 181.
- Check the **CXFS Client Log** in **CXFS Info** for warnings and errors regarding mounting filesystems.
- Check the cluster configuration to ensure that this node is configured to mount one or more filesystems.

Large Log Files on Windows

The `cxfs_client` service creates the following log file:

```
%ProgramFiles%\CXFS\log\cxfs_client.log
```

On an upgraded system, this log file may become quite large over a period of time if the verbosity level is increased. (New installations perform automatic log rotation when the file grows to 10 MB.)

To verify that log rotation is enabled, check the **Addition** arguments by modifying the installation (see "Modifying the CXFS Software for Windows" on page 168) and append the following if the `-z` option is not present:

```
-z 10000000
```

You must restart the `cxfs_client` service for the new settings to take effect. See "Start/Stop the `cxfs_client` Service for Windows" on page 164.

Windows Failure on Reboot

If the CXFS Windows node fails to start and terminates in a blue screen, reboot your computer and select the backup hardware profile with CXFS disabled. Alternatively, pressing `L` at the **Hardware Profile** menu will select the last configuration that was successfully started and shut down. If the node has only one hardware profile, press the spacebar after selecting the boot partition to get to the **Hardware Profile** menu.

`NO_MORE_SYSTEM_PTES` Error Message

A Windows problem may affect Windows CXFS nodes that are performing large asynchronous I/O operations. If the Windows node crashes with a `NO_MORE_SYSTEM_PTES` message, following these steps may help:

Caution: You should only try this if you are familiar with editing the registry and the risks involved in making these modifications. Doing so without proper experience may cause damage to your system. The value of `SystemPages` should only be increased above `110000` after consulting with a Microsoft Technical Support Engineer.

Edit the following `DWORD` registry values

- Set the value of `PagedPoolSize` to `0`:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PagedPoolSize
```

- Set the value of `SystemPages` to `110000`:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\SystemPages
```

Reboot the system to apply the changes.

Application Cannot Create File Under CXFS Drive Letter

If an application requires that it be able to create files and/or directories in the root of the CXFS drive, you must create a virtual drive for the system that maps to a mounted filesystem directory.

This can be performed using the `subst` command from the command prompt. For example, to use the CXFS filesystem `X:\mnt\tp9500_0` to the free drive letter `V`, you would enter the following:

```
C:\> subst V: X:\mnt\tp9500_0
```

To remove the mapping, run:

```
C:\> subst V: /D
```

Installation `File Not Found` Errors

Some installation programs are known to use old Windows APIs for file operations so that they work on older versions of Windows. These APIs use 8.3 filenames rather then the full filename, so the installation may fail with `file not found` or similar errors. In general, SGI recommends that you install software to a local disk and use CXFS filesystems primarily for data storage.

No WWPNs Detected for Windows

If no WWPNs are detected, there will be messages about loading the HBA/SNIA library logged to the `%ProgramFiles%\CXFS\log\cxfs_client.log` file.

If no WWPNs are detected, you must manually specify the WWPNs in the fencing file.

Note: This method does not work if the WWPNs are partially discovered.

The `%ProgramFiles%\CXFS\fencing.conf` file enumerates the WWPN for all of the HBAs that will be used to mount a CXFS filesystem. There must be a line for the HBA WWPN as a 64-bit hexadecimal number.

Note: The WWPN is that of the HBA itself, **not** any of the devices that are visible to that HBA in the fabric.

If used, `%ProgramFiles%\CXFS\fencing.conf` must contain a simple list of WWPNs, one per line. You must update it whenever the HBA configuration changes, including the replacement of an HBA.

This section discusses the following:

- "Determining the WWPN for a QLogic FC Switch" on page 189
- "Determining the WWPN for a Brocade FC Switch" on page 190

Determining the WWPN for a QLogic FC Switch

Do the following to determine the WWPN for a QLogic switch:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Connect to the switch and log in as user `admin`. (The password is `password` by default).
3. Enter the `show topology` command to retrieve the WWPN numbers.

   For example:

```
SANbox #> show topology

  Unique ID Key
  -------------
  A = ALPA,  D = Domain ID,  P = Port ID

  Port    Loc   Local                   Rem  Remote                  Unique
  Number  Type  PortWWN                 Type NodeWWN                 ID
  ------  ----  -------                 ---- -------                 ------
  0       F     20:00:00:c0:dd:06:ff:7f N    20:00:00:01:ff:03:05:b2 020000   P
  2       F     20:02:00:c0:dd:06:ff:7f N    20:01:00:e0:8b:32:ba:14 020200   P
  4       F     20:04:00:c0:dd:06:ff:7f N    20:00:00:01:ff:03:05:b2 020400   P
  5       F     20:05:00:c0:dd:06:ff:7f N    20:00:00:e0:8b:0b:81:24 020500   P
  6       F     20:06:00:c0:dd:06:ff:7f N    20:01:00:e0:8b:32:06:c8 020600   P
  8       F     20:08:00:c0:dd:06:ff:7f N    20:00:00:01:ff:03:05:b2 020800   P
  12      F     20:0c:00:c0:dd:06:ff:7f N    20:00:00:01:ff:03:05:b2 020c00   P
  15      F     20:0f:00:c0:dd:06:ff:7f N    20:00:00:e0:8b:10:04:13 020f00   P
  17      E     20:11:00:c0:dd:06:ff:7f E    10:00:00:c0:dd:06:fb:04 1(0x1) D
  19      E     20:13:00:c0:dd:06:ff:7f E    10:00:00:c0:dd:06:fb:04 1(0x1) D
```

The WWPN is the hexadecimal string in the `Remote NodeWWN` column are the numbers that you copy for the `fencing.conf` file. For example, the WWPN for port 0 is `20000001ff0305b2` (you must remove the colons from the WWPN reported in the `show topology` output in order to produce the string to be used in the fencing file).

4. Edit or create `%ProgramFiles%\CXFS\fencing.conf` and add the WWPN for the port. (Comment lines begin with #.)

   For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

   For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

   ```
   # WWPN of the HBA installed on this system
   #
   2000000173002c0b
   ```

5. To enable fencing, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Determining the WWPN for a Brocade FC Switch

Do the following to determine the WWPN for a Brocade switch:

1. Set up the switch and HBA. See the release notes for supported hardware.
2. Use the `telnet` command to connect to the switch and log in as user `admin`. (The password is `password` by default).
3. Execute the `switchshow` command to display the switches and their WWPN numbers.

   For example:

   ```
   brocade04:admin> switchshow
   switchName:     brocade04
   switchType:     2.4
   switchState:    Online
   switchRole:     Principal
   switchDomain:   6
   switchId:       fffc06
   switchWwn:      10:00:00:60:69:12:11:9e
   switchBeacon:   OFF
   port  0: sw  Online        F-Port  20:00:00:01:73:00:2c:0b
   port  1: cu  Online        F-Port  21:00:00:e0:8b:02:36:49
   port  2: cu  Online        F-Port  21:00:00:e0:8b:02:12:49
   port  3: sw  Online        F-Port  20:00:00:01:73:00:2d:3e
   port  4: cu  Online        F-Port  21:00:00:e0:8b:02:18:96
   ```

```
port  5: cu  Online        F-Port  21:00:00:e0:8b:00:90:8e
port  6: sw  Online        F-Port  20:00:00:01:73:00:3b:5f
port  7: sw  Online        F-Port  20:00:00:01:73:00:33:76
port  8: sw  Online        F-Port  21:00:00:e0:8b:01:d2:57
port  9: sw  Online        F-Port  21:00:00:e0:8b:01:0c:57
port 10: sw  Online        F-Port  20:08:00:a0:b8:0c:13:c9
port 11: sw  Online        F-Port  20:0a:00:a0:b8:0c:04:5a
port 12: sw  Online        F-Port  20:0c:00:a0:b8:0c:24:76
port 13: sw  Online        L-Port  1 public
port 14: sw  No_Light
port 15: cu  Online        F-Port  21:00:00:e0:8b:00:42:d8
```

The WWPN is the hexadecimal string to the right of the port number. For example, the WWPN for port 0 is `2000000173002c0b` (you must remove the colons from the WWPN reported in the `switchshow` output in order to produce the string to be used in the fencing file).

4. Edit or create `%ProgramFiles%\CXFS\fencing.conf` and add the WWPN for the port. (Comment lines begin with #.)

   For dual-ported HBAs, you must include the WWPNs of any ports that are used to access cluster disks. This may result in multiple WWPNs per HBA in the file; the numbers will probably differ by a single digit.

   For example, if you determined that port 0 is the port connected to the switch, your fencing file should contain the following:

   ```
   # WWPN of the HBA installed on this system
   #
   2000000173002c0b
   ```

5. To enable fencing, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Note: You may be able to use an HBA vendor-provided utility to determine the WWPN.

Unable to Join Multicast Group

If the Windows client is unable to join the multicast group, it may be that the CXFS client is starting before the network interface has been brought up. In this case, you

can delay the automatic start of the CXFS client. See "Delay Automatic Start of the CXFS Client (Windows Vista and Later)" on page 179.

Problems Specific to Windows Vista, Windows Server 2008, and Windows 7

This section discusses problems specific to Windows Vista and Windows Server 2008:

Node Loses Membership Due to Hibernation

If the Windows Vista, Windows Server 2008, or Windows 7 node hibernates, it will lose membership in the CXFS cluster. Hibernation is turned on by default for Windows Vista, Windows Server 2008, and Windows 7 and must be modified.

Do the following:

1. Select the following:

   Start
   > Control Panel
   > Power Options

2. Verify that **Put the computer to sleep** is set to **Never**.

Alternatively, you can use the following command:

```
C:\> powercfg -S SCHEME_MIN
```

Node Appears to be in Membership But Is Not

If the Windows Vista, Windows Server 2008, or Windows 7 node appears to be in membership when the `cxfs_info` command is run from the node but is not in membership according to administration tools run on a server-capable administration node, it may be that User Account Control is still enabled (it is enabled by default).

User Account Control is not appropriate for use with CXFS, and you must disable it. See step 4 in "Client Software Installation for Windows" on page 136.

Node Unable to Change Directory to a Mounted Filesystem

If you are unable to use the `cd` command on a Windows Vista, Windows Server 2008, or Windows 7 node for a filesystem that appears to be mounted, it may be that User Account Control is still enabled (it is enabled by default). For example, using a `cygwin` shell:

```
user@host /home/user
$ cd /cygdrive/x/mnt/stripefs
  -bash: cd: /cygdrive/x/mnt/stripefs: Input/Output error**
```

User Account Control is not appropriate for use with CXFS, and you must disable it. See step 4 in "Client Software Installation for Windows" on page 136.

Slow Installation

If the installation of the Windows Vista, Windows Server 2008, or Windows 7 operating system seems to take a long time or does not complete, it may be caused by the HBAs or SAN fabric.

To resolve this problem, do the following:

1. Disconnect the system from the SAN fabric.
2. Remove the HBAs from the system or disable them in the BIOS.
3. Install the operating system.
4. Reinstall or reenable the HBA.
5. Install CXFS.
6. Reconnect the SAN fabric.

Reporting Windows Problems

This section discusses the following:

- "Generating a Crash Dump on a Hung Windows Node" on page 195

Retaining Windows Information

To report problems about a Windows node, you should retain platform-specific information and save crash dumps.

When reporting a problem about a CXFS Windows node to SGI, run the following:

Start
> Program Files
> CXFS
> CXFS Dump

This will collect the following information:

- System information
- CXFS registry settings
- CXFS client logs
- CXFS version information
- Network settings
- Event log
- *(optionally)* Windows crash dump, as described in "Saving Crash Dumps for Windows" on page 195

In the dialog window, you will specify the location of the folder in which the `cxfsdump` output will be placed. The output will be placed beneath this folder, in a new folder whose name is of the form `CxfsDump_`*`date_time`*, where *date* is the numeric date (such as `20080925` for September 25, 2008) and *time* is in military notation to the nearest second (such as `214456` for 9:44pm, 56 seconds).

Inside the `CxfsDump_`*`date_time`* folder will be a collection of `log` and `txt` files. You should compress the folder and files (using `zip` or `tar`) and send them to SGI.

The `cxfsdump /?` command displays a help message.

You should also obtain information about the entire cluster by running the `cxfsdump` utility on a server-capable administration node. See the information in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Saving Crash Dumps for Windows

If you are experiencing crashes or if the Windows node hangs, you should configure the Windows node to save crash dumps to a filesystem that is not a CXFS filesystem. This crash dump can then be analyzed by SGI.

Do the following:

1. Click the right mouse button on the **My Computer** icon and select the following:

   Properties
   > Advanced
   > Startup and Recovery
   > Write debugging information to

2. Enter a path on a filesystem other than a CXFS filesystem. You may also select a **Kernel Memory Dump**, which is a smaller dump that typically contains enough information regarding CXFS problems.

3. Reboot the system to apply the changes.

Saving Application Crash Dumps for Windows Vista, Windows Server 2008, and Windows 7

When a user space application crashes, it will remain in the TaskManager. In the dialog pop up that appears, detailing the crash information, you should right-click the application that caused the crash and select the crash dump option. This will save the dump to the current **User** directory so that the dump can then be analyzed.

Note: If you close the dialog without saving, the process will be removed from the TaskManager and the dump information will be lost.

For more information, see the following Microsoft article:

http://support.microsoft.com/kb/931673

Generating a Crash Dump on a Hung Windows Node

If user applications on a Windows node are no longer responsive and cannot be killed, you should attempt to generate a crash dump by forcing the node to crash. After configuring the crash dump location (see "Saving Crash Dumps for Windows"

on page 195), you can modify the registry so that a combination of key strokes will cause the Windows node to crash.

To generate a crash dump, add the following new DWORD to the registry with a value of 1:

- USB keyboard:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\kbdhid\Parameters\CrashOnCtrlScroll
```

- PS/2 keyboard:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\i8042prt\Parameters\CrashOnCtrlScroll
```

Reboot the system to apply the changes.

To generate a crash on the node after applying these changes, hold the right CTRL key and press SCROLL LOCK twice. See the following for more information:

http://support.microsoft.com/?kbid=244139

Configuring Client-Only Nodes

This chapter provides an overview of the procedures to add the client-only nodes to an established cluster. It assumes that you already have a cluster of server-capable administration nodes installed and running with mounted filesystems. These procedures will be performed by you or by SGI service personnel.

All CXFS administrative tasks other than restarting the Windows node must be performed using the CXFS GUI (invoked by the `cxfsmgr` command and connected to a server-capable administration node) or the `cxfs_admin` command on any host that has access permission to the cluster. The GUI and `cxfs_admin` provide a guided configuration and setup help for defining a cluster.

This section discusses the following tasks in cluster configuration:

For detailed configuration instructions, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Defining the Client-Only Nodes

To add a client-only node to a CXFS cluster, you must define it as a node in the pool.

Do the following to determine the value for the hostname field in the GUI:

- Linux: use the value displayed by `/bin/hostname`
- Mac OS X: use the value displayed by `/bin/hostname`
- Windows: select the following:

  Start
  > Settings
  > Network and Dial-up Connections
  > Advanced
  > Network Identification

For example, the following shows the entries used to define a node named `mac1` in the `mycluster` cluster:

```
# /usr/cluster/bin/cxfs_admin -A -i mycluster
cxfs_admin:mycluster> create node name=mac1 os=macosx private_net=192.168.0.178
Event at [ Jan 21 15:58:02 ]
Node "mac1" has been created, waiting for it to join the cluster...
Waiting for node mac1, current status: Inactive
Waiting for node mac1, current status: Establishing membership
Waiting for node mac1, current status: Probing XVM volumes
Operation completed successfully
```

Or, in prompting mode:

```
# /usr/cluster/bin/cxfs_admin -i mycluster
Event at [ Jan 21 15:59:02 ]
cxfs_admin:mycluster> create node
  Specify the attributes for create node:
   name? mac1
   type? client_only
   os? macosx
   private_net? 192.168.0.178
Event at [ Jan 21 15:59:10 ]
Node "man1" has been created, waiting for it to join the cluster...
Waiting for node mac1, current status: Inactive
Waiting for node mac1, current status: Establishing membership
Waiting for node mac1, current status: Probing XVM volumes
Operation completed successfully
```

For client-only nodes, you must specify a unique node ID if you use the GUI; `cxfs_admin` provides a default node ID.

For details about these commands, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Adding the Client-Only Nodes to the Cluster (GUI)

If you are using the GUI, you must add the defined nodes to the cluster. This happens by default if you are using `cxfs_admin`.

Depending upon your filesystem configuration, you may also need to add the node to the list of clients that have access to the volume. See "Mounting Filesystems" on page 202.

Defining the Switch for I/O Fencing

I order to protect data integrity, you must use I/O fencing on client-only nodes (or reset for those nodes with system controllers). I/O fencing requires a switch; see the release notes for supported switches.

For example, for a QLogic switch named `myswitch`:

```
cxfs_admin:mycluster> create switch name=myswitch vendor=qlogic
```

After you have defined the switch, you must ensure that all of the switch ports that are connected to the cluster nodes are enabled. To determine port status, enter the following on a server-capable administration node:

```
server-admin# /usr/cluster/bin/hafence -v
```

If there are disabled ports that are connected to cluster nodes, you must enable them. Log into the switch as user `admin` and use the following command:

```
switch# portEnable portnumber
```

You must then update the switch port information

For example, suppose that you have a cluster with port 0 connected to the node `blue`, port 1 connected to the node `green`, and port 5 connected to the node `yellow`, all of which are defined in cluster `colors`. The following output shows that the status of port 0 and port 1 is `disabled` and that the host is `UNKNOWN` (as opposed to port 5, which has a status of `enabled` and a host of `yellow`). Ports 2, 3, 4, 6, and 7 are not connected to nodes in the cluster and therefore their status does not matter.

```
server-admin# /usr/cluster/bin/hafence -v
  Switch[0] "ptg-brocade" has 8 ports
    Port 0 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
    Port 1 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
    Port 2 type=FABRIC status=enabled  hba=210000e08b05fecf on host UNKNOWN
    Port 3 type=FABRIC status=enabled  hba=210000e08b01fec5 on host UNKNOWN
    Port 4 type=FABRIC status=enabled  hba=210000e08b01fec3 on host UNKNOWN
    Port 5 type=FABRIC status=enabled  hba=210000e08b019ef0 on host yellow
    Port 6 type=FABRIC status=enabled  hba=210000e08b0113ce on host UNKNOWN
    Port 7 type=FABRIC status=enabled  hba=210000e08b027795 on host UNKNOWN
```

In this case, you would need to enable ports 0 and 1:

Logged in to the switch:

```
switch# portEnable 0
switch# portEnable 1
```

Logged in to a server-capable administration node:

```
server-admin# /usr/cluster/bin/hafence -v
  Switch[0] "ptg-brocade" has 8 ports
    Port 0 type=FABRIC status=disabled hba=210000e08b0103b8 on host UNKNOWN
    Port 1 type=FABRIC status=disabled hba=210000e08b0102c6 on host UNKNOWN
    Port 2 type=FABRIC status=enabled  hba=210000e08b05fecf on host UNKNOWN
    Port 3 type=FABRIC status=enabled  hba=210000e08b01fec5 on host UNKNOWN
    Port 4 type=FABRIC status=enabled  hba=210000e08b01fec3 on host UNKNOWN
    Port 5 type=FABRIC status=enabled  hba=210000e08b019ef0 on host yellow
    Port 6 type=FABRIC status=enabled  hba=210000e08b0113ce on host UNKNOWN
    Port 7 type=FABRIC status=enabled  hba=210000e08b027795 on host UNKNOWN

server-admin# /usr/cluster/bin/hafence -v
  Switch[0] "ptg-brocade" has 8 ports
    Port 0 type=FABRIC status=disabled hba=210000e08b0103b8 on host blue
    Port 1 type=FABRIC status=disabled hba=210000e08b0102c6 on host green
    Port 2 type=FABRIC status=enabled  hba=210000e08b05fecf on host UNKNOWN
    Port 3 type=FABRIC status=enabled  hba=210000e08b01fec5 on host UNKNOWN
    Port 4 type=FABRIC status=enabled  hba=210000e08b01fec3 on host UNKNOWN
    Port 5 type=FABRIC status=enabled  hba=210000e08b019ef0 on host yellow
    Port 6 type=FABRIC status=enabled  hba=210000e08b0113ce on host UNKNOWN
    Port 7 type=FABRIC status=enabled  hba=210000e08b027795 on host UNKNOWN
```

Starting CXFS Services (GUI)

After adding the client-only nodes to the cluster with the GUI, you must start CXFS services for them, which enables the node by setting a flag for the node in the cluster database. This happens by default with `cxfs_admin`.

Verifying LUN Masking

You should verify that the HBA has logical unit (LUN) masking configured such that the LUNs are visible to all the nodes in the cluster after you connect the HBA to the switch and before configuring the filesystems with XVM. For more information, see the RAID documentation.

Mounting Filesystems

If you have specified that the filesystems are to be automatically mounted on any newly added nodes (such as setting `mount_new_nodes=true` for a filesystem in `cxfs_admin`), you do not need to specifically mount the filesystems on the new client-only nodes that you added to the cluster.

If you have specified that filesystems **will not be automatically mounted** (for example, by setting the advanced-mode `mount_new_nodes=false` for a filesystem in `cxfs_admin`), you can do the following to mount the new filesystem:

- With `cxfs_admin`, use the following command to mount the specified filesystem:

  `mount` *filesystemname* `nodes=`*nodename*

  For example:

  ```
  cxfs_admin:mycluster> mount fs1 nodes=mac2
  ```

  You can leave `mount_new_nodes=false`. You do not have to unmount the entire filesystem.

- With the GUI, you can mount the filesystems on the new client-only nodes by unmounting the currently active filesystems, enabling the mount on the required nodes, and then performing the actual mount.

Note: SGI recommends that you enable the *forced unmount* feature for CXFS filesystems, which is turned off by default; see:

- "Enable Forced Unmount When Appropriate" on page 16
- "Forcing Unmount of CXFS Filesystems" on page 203

Unmounting Filesystems

You can unmount a filesystem from all nodes in the cluster or from just the node you specify.

For example, to unmount the filesystem `fs1` from all nodes:

```
cxfs_admin:mycluster> unmount fs1
```

To unmount the filesystem only from the node `mynode`:

```
cxfs_admin:mycluster> unmount fs1 nodes=mynode
```

Forcing Unmount of CXFS Filesystems

Normally, an unmount operation will fail if any process has an open file on the filesystem. However, a *forced unmount* allows the unmount to proceed regardless of whether the filesystem is still in use.

For example:

```
cxfs_admin:mycluster> create filesystem name=myfs forced_unmount=true
```

Using the CXFS GUI, define or modify the filesystem to unmount with force and then unmount the filesystem.

For details, see the “CXFS Filesystems Tasks with the GUI” sections of the GUI chapter in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Restarting the Windows Node

After completing the steps in "Postinstallation Steps for Windows" on page 144 and this chapter, you should restart the Windows node. This will automatically start the driver and the CXFS Client service.

When you log into the node after restarting it, Windows Explorer will list the CXFS drive letter, which will contain the CXFS filesystems configured for this node.

Verifying the Cluster Configuration

To verify that the client-only nodes have been properly added to the cluster, run the `cxfs-config` command on the metadata server. For example:

```
/usr/cluster/bin/cxfs-config -all -check
```

This command will dump the current cluster nodes, private network configuration, filesystems, XVM volumes, failover hierarchy, and switches. It will check the configuration and report any common errors. You should rectify these error before starting CXFS services.

Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)

To verify general connectivity in a multicast environment (other than on a Windows node), you can execute a `ping` command on the `224.0.0.1` IP address.

To verify the CXFS heartbeat, use the `224.0.0.250` IP address. The `224.0.0.250` address is the default CXFS heartbeat multicast address (because it is the default, this address does not have to appear in the `/etc/hosts` file).

Note: A node is capable of responding only when the administration daemons (`fs2d`, `cmond`, `cad`, and `crsd`) or the `cxfs_client` daemon is running.

For example, to see the response for two packets sent from Linux IP address `10.0.0.27` to the multicast address for CXFS heartbeat and ignore loopback, enter the following:

```
linux# ping -I 10.0.0.27 -L -c 2 224.0.0.250
```

Note: By default, most Linux nodes will not respond to a multicast `ping`. To enable multicast `ping` for Linux nodes, see "Verifying the Private and Public Networks for Linux" on page 39.

To override the default address, you can use the `-c` and `-m` options or make the name `cluster_mcast` resolvable on all nodes (such as in the `/etc/hosts` file). For more information, see the `cxfs_client` man page.

Verifying the Cluster Status

To verify that the client-only nodes have been properly added to the cluster and that filesystems have been mounted, use the view area of the CXFS GUI, the `cxfs_admin status` command, or the `clconf_info` command (on a server-capable administration node) and the `cxfs_info` command (on a client-only node).

For example, using `cxfs_admin`:

```
cxfs_admin:clusterOne> status
Event at [ Oct 22 13:40:51 ]
Cluster         : clusterOne
Tiebreaker      :
Client Licenses : enterprise   allocated 0 of 256
                  workstation  allocated 2 of 50
------------------  --------  --------  ----------------------------------------
Node                Cell ID   Age       Status
------------------  --------  --------  ----------------------------------------
bert *              0         8         Mounted 1 of 2 filesystems
cxfsxe5 *           1         7         Stable
twig *              2         -         Inactive
cxfsxe10            3         -         Disabled
penguin17           4         0         Establishing membership
pg-27               5         0         Establishing membership

------------------  ------------------  ----------------------------------------
Filesystem          Server              Status
------------------  ------------------  ----------------------------------------
zj01s0              cxfsxe5             1 of 6 nodes mounted, bert trying to mount
zj01s1              bert                Mounted [2 of 6 nodes]

------------------  ----------  ------------------------------------------------
Switch              Port Count  Known Fenced Ports
------------------  ----------  ------------------------------------------------
brocade26cp1        192         4, 20, 21, 132, 223
```

The following example for a different cluster shows `clconf_info` output:

```
cxfsxe5:~ # /usr/cluster/bin/clconf_info

Event at [2009-10-22 13:41:24]

Membership since Thu Oct 22 13:39:22 2009
_________________ ______ ________ ______ ______
Node              NodeID Status   Age    CellID
_________________ ______ ________ ______ ______
cxfsxe5                1 up            7      1
twig                   2 DOWN          -      2
bert                   3 DOWN          -      0
pg-27                  4 DOWN          -      5
penguin17              5 DOWN          -      4
cxfsxe10               6 inactive      -      3
_________________ ______ ________ ______ ______
2 CXFS FileSystems
/dev/cxvm/zj01s1 on /mnt/zj01s1  enabled  server=(cxfsxe5)  0 client(s)=()  status=UP
/dev/cxvm/zj01s0 on /mnt/zj01s0  enabled  server=(cxfsxe5)  0 client(s)=()  status=UP
```

On client-only nodes, the `cxfs_info` command serves a similar purpose. The command path is as follows:

- Linux and Mac OS X: `/usr/cluster/bin/cxfs_info`
- Windows: `%ProgramFiles%\CXFS\cxfs_info.exe`

On Linux and Mac OS X, you can use the `-e` option to wait for events, which keeps the command running until you kill the process and the `-c` option to clear the screen between updates.

For example, on a Linux node named `pg-27`:

```
pg-27% /usr/cluster/bin/cxfs_info
cxfs_client status [timestamp Oct 22 13:40:46 / generation 5648504]

CXFS client:
    state: reconfigure (5), cms: quiesce, xvm: down, fs: down
Cluster:
    clusterOne (9) - enabled
Local:
    pg-27 (5) - enabled
```

```
Servers:
    bert       enabled  DOWN  0
    cxfsxe5    enabled  DOWN  1
    twig       enabled  DOWN  2
Nodes:
    cxfsxe10   disabled DOWN  3
    penguin17  enabled  DOWN  4
    pg-27      enabled  DOWN  5
Filesystems:
    zj01s0
    zj01s1
```

The `CXFS client` line shows the state of the client in the cluster, which can be one of the following states:

`bootstrap`	Initial state after starting `cxfs_client`, while listening for bootstrap packets from the cluster.
`connect`	Connecting to the CXFS metadata server.
`query`	The client is downloading the cluster database from the metadata server.
`reconfigure`	The cluster database has changed, so the client is reconfiguring itself to match the cluster database.
`stable`	The client has been configured according to what is in the cluster database.
`stuck`	The client is unable to proceed, usually due to a configuration error. Because the problem may be transient, the client periodically reevaluates the situation. The number in parenthesis indicates the number of seconds the client will wait before retrying the operation. With each retry, the number of seconds to wait is increased; therefore, the higher the number the longer it has been stuck. See the log file for more information.
`terminate`	The client is shutting down.

The `cms` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	The client is not in membership.
`fetal`	The client is joining membership.
`up`	The client is in membership.
`quiesce`	The client is dropping out of membership.

The `xvm` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	After membership, but before any XVM information has been gathered.
`fetal`	Gathering XVM information.
`up`	XVM volumes have been retrieved.

The `fs` field has the following states:

`unknown`	Initial state before connecting to the metadata server.
`down`	One or more filesystems are not in the desired state.
`up`	All filesystems are in the desired state.
`retry`	One or more filesystems cannot be mounted/unmounted, and will retry. See the "Filesystem" section of `cxfs_info` output to see the affected filesystems.

Verifying the I/O Fencing Configuration

To determine if a node is correctly configured for I/O fencing, log in to a server-capable administration node and use the `cxfs-config`(8) command. For example:

```
server-admin# /usr/cluster/bin/cxfs-config
```

The failure hierarchy for a client-only node should be listed as `Fence, Shutdown`, as in the following example:

```
Machines:
     node cxfswin2: node 102   cell 1  enabled  Windows client_only
         hostname: cxfswin2.melbourne.sgi.com
         fail policy: Fence, Shutdown
         nic 0: address: 192.168.0.102 priority: 1
```

See "Defining the Client-Only Nodes" on page 198 to change the failure hierarchy for the node if required.

The HBA ports should also be listed in the switch configuration:

```
Switches:
     switch 1: 16 port brocade admin@asg-fcsw7 <no ports masked>
         port 5: 210200e08b51fd49 cxfswin2
         port 15: 210100e08b32d914 admin1
     switch 2: 16 port brocade admin@asg-fcsw8 <no ports masked>
         port 5: 210300e08b71fd49 cxfswin2
         port 14: 210000e08b12d914 admin1
```

No warnings or errors should be displayed regarding the failure hierarchy or switch configuration.

If the HBA ports for the client node are not listed, see the following:

- "Configuring I/O Fencing for Linux" on page 45
- "Configuring I/O Fencing for Mac OS X" on page 88
- "Configuring I/O Fencing for Windows (FC Only)" on page 163

Verifying Access to XVM Volumes

To verify that a client node has access to all XVM volumes that are required to mount the configured filesystems, log on to a server-capable administration node and run:

```
server-admin# /usr/cluster/bin/cxfs-config -xvm
```

This will display the list of filesystems and the XVM volume and volume elements used to construct those filesystems. For example:

```
 fs stripe1: /mnt/stripe1         enabled
     device = /dev/cxvm/stripe1
     force = false
     options = []
     servers = cxfs5 (0), cxfs4 (1)
     clients = cxfs4, cxfs5, cxfs6, cxfsmac4, cxfssun1
     xvm:
         vol/stripe1                      0 online,open
             subvol/stripe1/data      2292668416 online,open
                 stripe/stripe1           2292668416 online,open
                     slice/d9400_0s0          1146334816 online,open
                     slice/d9400_1s0          1146334816 online,open

         data size: 1.07 TB
```

You can then run the `xvm` command to identify the XVM volumes and disk devices. This provides enough information to identify the device's WWN, LUN, and controller. In the following example, the `slice/d9400_0s0` from `phys/d9400_0` is LUN 0 located on a RAID controller with WWN `200500a0b80cedb3`.

```
server-admin# /sbin/xvm show -e -t vol
vol/stripe1                      0 online,open
     subvol/stripe1/data      2292668416 online,open
         stripe/stripe1           2292668416 online,open (unit size: 1024)
             slice/d9400_0s0          1146334816 online,open (d9400_0:/dev/rdsk/200500a0b80cedb3/lun0vol/c2p1)
             slice/d9400_1s0          1146334816 online,open (d9400_1:/dev/rdsk/200400a0b80cedb3/lun1vol/c3p1)
```

On Linux and Mac OS X platforms, you can then run the xvm command on the client to identify the matching disk devices on the client. For example:

```
linux# /sbin/xvm show -e -t vol
```

Note: The xvm command on the Windows does not display WWNs.

If a disk device has not been found for a particular volume element, the following message will be displayed instead of the device name:

```
no direct attachment on this cell
```

Using the device information from the server-capable administration node, it should then be possible to determine if the client can see the same devices using the client HBA tools and the RAID configuration tool.

To see the complete list of volumes and devices mappings, especially when XVM failover V2 is configured, run:

```
linux# /sbin/xvm show -v phys
```

For more information about xvm, see the *XVM Volume Manager Administrator Guide*.

General Troubleshooting

This chapter contains the following:

Also see:

For more advanced cluster troubleshooting, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage.*

Identifying Problems

This section provides tips about identifying problems:

Is the Node Configured Correctly?

To determine the current configuration of a node in a cluster, run the following command on a CXFS server-capable administration node:

```
server-admin# /usr/cluster/bin/cxfs-config -all
```

For more information, see "Verifying the Cluster Status" on page 205.

Confirm that the host type, private network, and failure hierarchy are configured correctly, and that no warnings or errors are reported. You should rectify any warnings or errors before proceeding with further troubleshooting.

Is the Node in Membership?

To determine if the node is in the cluster membership, use the tools described in "Verifying the Cluster Status" on page 205.

If the client is not in membership, see:

- "Verifying the Cluster Configuration" on page 204
- "Verifying Connectivity in a Multicast Environment (Linux and Mac OS X Nodes)" on page 204
- "Unable to Achieve Membership" on page 219

Is the Node Is Fenced?

To determine if a client-only node is fenced, log in to a CXFS server-capable administration node and use the `hafence`(1m) command. A fenced port is displayed as `status=disabled`.

In the following example, all ports that have been registered as CXFS host ports are not fenced:

```
admin# /usr/cluster/bin/hafence -q
Switch[0] "brocade04" has 16 ports
Port 4 type=FABRIC status=enabled hba=210000e08b0042d8 on host o200c
Port 5 type=FABRIC status=enabled hba=210000e08b00908e on host cxfs30
Port 9 type=FABRIC status=enabled hba=2000000173002d3e on host cxfssun3
```

All switch ports can also be shown with `hafence`:

```
admin# /usr/cluster/bin/hafence -v
Switch[0] "brocade04" has 16 ports
Port 0 type=FABRIC status=enabled hba=2000000173003b5f on host UNKNOWN
Port 1 type=FABRIC status=enabled hba=2000000173003adf on host UNKNOWN
Port 2 type=FABRIC status=enabled hba=210000e08b023649 on host UNKNOWN
Port 3 type=FABRIC status=enabled hba=210000e08b021249 on host UNKNOWN
Port 4 type=FABRIC status=enabled hba=210000e08b0042d8 on host o200c
Port 5 type=FABRIC status=enabled hba=210000e08b00908e on host cxfs30
Port 6 type=FABRIC status=enabled hba=2000000173002d2a on host UNKNOWN
Port 7 type=FABRIC status=enabled hba=2000000173003376 on host UNKNOWN
Port 8 type=FABRIC status=enabled hba=2000000173002c0b on host UNKNOWN
Port 9 type=FABRIC status=enabled hba=2000000173002d3e on host cxfssun3
Port 10 type=FABRIC status=enabled hba=2000000173003430 on host UNKNOWN
Port 11 type=FABRIC status=enabled hba=200900a0b80c13c9 on host UNKNOWN
Port 12 type=FABRIC status=disabled hba=0000000000000000 on host UNKNOWN
Port 13 type=FABRIC status=enabled hba=200d00a0b80c2476 on host UNKNOWN
Port 14 type=FABRIC status=enabled hba=1000006069201e5b on host UNKNOWN
Port 15 type=FABRIC status=enabled hba=1000006069201e5b on host UNKNOWN
```

When the client-only node joins membership, any fences on any switch ports connected to that node should be lowered and the status changed to `enabled`.

However, if the node still does not have access to the storage, do the following:

- Check that the HBA WWPNs were correctly identified. See "Verifying the I/O Fencing Configuration" on page 208.
- Check the `cxfs_client` log file for warnings or errors while trying to determine the HBA WWPNs. See "No HBA WWPNs are Detected" on page 221.
- Log in to the Fibre Channel, SAS, or InfiniBand switch. Check the status of the switch ports and confirm that the WWPNs match those identified by `cxfs_client`.

Is the Node Mounting All Filesystems?

To determine if the node has mounted all configured filesystems, use the tools described in "Verifying the Cluster Status" on page 205.

If the client has not mounted all filesystems, see:

- "Verifying the Cluster Configuration" on page 204
- "Verifying Access to XVM Volumes" on page 210
- "Is the Node Is Fenced?" on page 214
- Appendix C, "Mount Options Support" on page 229

Can the Node Access All Filesystems?

To determine if the client-only node can access a filesystem, navigate the filesystem and attempt to create a file.

If the filesystem appears to be empty, the mount may have failed or been lost. See:

- "Is the Node Is Fenced?" on page 214
- "Verifying Access to XVM Volumes" on page 210

If accessing the filesystem hangs the viewing process, see "Filesystem Appears to Be Hung" on page 220.

Are There Error Messages?

When determining the state of the client-only node, you should check error message logs to help identify any problems.

Appendix A, "Operating System Path Differences" on page 223 lists the location of the `cxfs_client` log file for each platform. This log is also displayed in the Windows version of `cxfs_info`.

Each platform also has its own system log for kernel error messages that may also capture CXFS messages. See:

- "Log Files on Linux" on page 28
- "Log Files on Mac OS X" on page 68

• "Log Files and Cluster Status for Windows" on page 106

There are various logs located on the CXFS server-capable administration nodes. For more information, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Note: The `syslog` file or Linux `/var/log/messages` file may contain spurious error messages. This problem occurs in clusters with multiple private networks when one of the active network interfaces is downed using the `ifconfig` command. This problem may also happen when the network is interrupted for other reasons.

Some of the transport failure messages generated may have cell IDs that are different from the cell ID of the node with the downed interface. The spurious error messages do not appear to affect the continued operation of the cluster.

What Is the Network Status?

Use the `netstat` command on a client-only node to determine the network status.

For example, to determine if you have a bad connection, you could enter the following from a DOS console on the Windows platform:

```
C:\Documents and Settings\cxfsqa>netstat -e -s
```

The `-s` option shows per-protocol statistics. The Linux and Windows systems support the `-e` option, which shows Ethernet statistics. See the `netstat`(1) man page for information about options.

What Is the Status of XVM Mirror Licenses?

To view the current status of XVM mirror licenses, use the following command and search for the line containing the keyword `mirrors`:

```
xvm show -subsystem
```

For example:

```
# xvm show -subsystem
XVM Subsystem Information:
--------------------------
apivers:                26
config gen:             33
privileged:             1
clustered:              1
cluster initialized:    1
user license enabled:   1
local mirrors enabled:  1
cluster mirrors enabled: 1
snapshot enabled:       1
snapshot max blocks:    -1
snapshot blocks used:   0
```

Potential Problems and Solutions

This section contains the following typical problems that apply to any platform:

cdb Error in the cxfs_client Log

The following errors in the cxfs_client may log indicate that the client is not found in the cluster database:

```
cxfs_client: cis_client_run querying CIS server
cxfs_client: cis_cdb_go ERROR: Error returned from server: cdb error (6)
```

Run the `cxfs-config` command on the metadata server and verify that the client's hostname appears in the cluster database. For additional information about the error, review the `/var/cluster/log/fs2d_log` file on the metadata server.

Unable to Achieve Membership

If `cxfs_info` does not report that CMS is `UP`, do the following:

1. Check that `cxfs_client` is running. See:
   - "Start/Stop `cxfs_client` for Linux" on page 45
   - "Start/Stop `cxfs_client` for Mac OS X" on page 88
   - "Start/Stop the `cxfs_client` Service for Windows" on page 164
2. Look for other warnings and error messages in the `cxfs_client` log file. For the location of the log file on different platforms, see Appendix A, "Operating System Path Differences" on page 223.
3. Check `cxfs-config` output on the CXFS server-capable administration node to ensure that the client is correctly configured and is reachable via the configured CXFS private network. For example:

   ```
   server-admin# /usr/cluster/bin/cxfs-config -all
   ```

4. Check that the client is enabled into the cluster by running `clconf_info` on a CXFS server-capable administration node.
5. Look in the system log on the CXFS metadata server to ensure that the server detected the client that is attempting to join membership and check for any other CXFS warnings or errors.
6. Check that the metadata server has the node correctly configured in its hostname lookup scheme (`/etc/host` file or DNS).
7. If you are still unable to resolve the problem, reboot the client node.
8. If rebooting the client node in step 7 did not resolve the problem, restart the cluster administration daemons (`fs2d`, `cad`, `cmond`, and `crsd`) on the metadata server. This step may result in a temporary delay in access to the filesystem from all nodes.

9. If restarting cluster administration daemons in step 8 did not solve the problem, reboot the metadata server. This step may result in the filesystems being unmounted on all nodes.

Filesystem Appears to Be Hung

If any CXFS filesystem activity appears to be hung, do the following:

1. Check that the client is still in membership and that the filesystem is mounted according to `cxfs_info`.
2. Check on the metadata server to see if any messages are more than a few seconds in age (known as a *stuck message*).
3. If there is a stuck message, gather information for SGI support:
   - Find the stack trace for the stuck thread. For example:

```
crash> bt 0xe00000305f2a0000
 #0 [BSP:e00000305f2a1e90] schedule at a0000001006e3d30
 #1 [BSP:e00000305f2a1e60] schedule_timeout at a0000001006e4460
 #2 [BSP:e00000305f2a1e20] __down at a0000001006e5d60
 #3 [BSP:e00000305f2a1df0] down at a0000001000d5db0
 #4 [BSP:e00000305f2a1dd0] xfs_buf_lock at a000000203e15030
 #5 [BSP:e00000305f2a1d80] _xfs_buf_find at a000000203e18cf0
 #6 [BSP:e00000305f2a1d20] xfs_buf_get_flags at a000000203e18f60
 #7 [BSP:e00000305f2a1ce8] xfs_buf_read_flags at a000000203e19220
 #8 [BSP:e00000305f2a1c90] xfs_trans_read_buf at a000000203df6750
 #9 [BSP:e00000305f2a1c48] xfs_btree_read_bufs at a000000203d88690
#10 [BSP:e00000305f2a1ba8] xfs_inobt_lookup at a000000203db79b0
#11 [BSP:e00000305f2a1b68] xfs_inobt_lookup_eq at a000000203db8010
#12 [BSP:e00000305f2a1ab0] xfs_dialloc at a000000203db4ca0
#13 [BSP:e00000305f2a1a38] xfs_ialloc at a000000203dc6660
#14 [BSP:e00000305f2a1990] xfs_dir_ialloc at a000000203df90e0
#15 [BSP:e00000305f2a18f8] xfs_mkdir at a000000203e076e0
#16 [BSP:e00000305f2a18a8] cxfs_mkdir at a000000203c0a460
#17 [BSP:e00000305f2a1858] dmapi_bnc_mkdir at a000000203bb6b60
#18 [BSP:e00000305f2a17f0] bhvlock_vop_mkdir at a000000203bb0330
#19 [BSP:e00000305f2a1788] xfs_vn_mknod at a000000203e23110
#20 [BSP:e00000305f2a1758] xfs_vn_mkdir at a000000203e23460
#21 [BSP:e00000305f2a1710] vfs_mkdir at a0000001001cb320
#22 [BSP:e00000305f2a16b8] cxfs_server_lock_mkdir at a000000203c21980
```

```
#23 [BSP:e00000305f2a1438] I_dsvn_create_0 at a000000203bea810
#24 [BSP:e00000305f2a1398] dsvn_msg_dispatcher at a000000203b1e4a0
#25 [BSP:e00000305f2a1330] mesg_demux at a00000020393e0a0
#26 [BSP:e00000305f2a11f0] mtcp_notify at a000000203954200
#27 [BSP:e00000305f2a1148] tsv_thread_setup at a000000203a2b8e0
#28 [BSP:e00000305f2a10c8] kthread_init at a000000203a22050
#29 [BSP:e00000305f2a10a0] kernel_thread_helper at a000000100014a30
#30 [BSP:e00000305f2a10a0] start_kernel_thread at a00000010000a4c0
```

- Run `cxfsdump` on the metadata server.
- Run `cxfsdump` on the client that has the stuck message.
- If possible, force the client that has the stuck message to generate a crash dump.

4. Reboot the client that has the stuck message. This is required for CXFS to recover.

No HBA WWPNs are Detected

On most platforms, the `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) in the system that are connected to a switch that is configured in the cluster database. These HBAs will then be available for fencing.

However, if no WWPNs are detected, there will be messages about loading the HBA/SNIA library.

See:

- "Configuring I/O Fencing for Linux" on page 45
- "Configuring I/O Fencing for Mac OS X" on page 88
- "Configuring I/O Fencing for Windows (FC Only)" on page 163

Membership Is Prevented by Firewalls

If a client has trouble obtaining membership, verify that the system firewall is configured for CXFS use. See "Configure Firewalls for CXFS Use" on page 17.

Devices are Unknown

You can run the `cxfs-reprobe` script on a client-only node (other than Windows) to look for devices and perform a SCSI bus reset if necessary. `cxfs-reprobe` will also issue an XVM probe to tell XVM that there may be new devices available:

```
client# /var/cluster/cxfs_client-scripts/cxfs-reprobe
```

Clients Cannot Join the Cluster After Relocation

If a CXFS client fails or exits the cluster during the metadata server relocation process, the relocation process and the client recovery are likely to hang. This prevents any clients, including the failed client, from joining the cluster.

Once in this state, it may be possible to resolve the deadlock by resetting or power-cycling the `fs2d` quorum master. To determine the quorum master, see the instructions in *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

Using SGI Knowledgebase

If you encounter problems and have an SGI support contract, see:

https://support.sgi.com

If you need further assistance, contact SGI Support.

Reporting Problems to SGI

When reporting a problem with a client-only node, it is important to retain the appropriate information; having access to this information will greatly assist SGI in the process of diagnosing and fixing problems. The methods used to collect required information for problem reports are platform-specific:

- "Reporting Linux Problems" on page 63
- "Reporting Mac OS X Problems" on page 100
- "Reporting Windows Problems" on page 193

Appendix A

Operating System Path Differences

This appendix lists the location of CXFS-specific commands and files. For more information, see the `cxfs_client` man page.

Table A-1 Linux Paths

Component	Path
CXFS client service:	`/usr/cluster/bin/cxfs_client`
CXFS status:	`/usr/cluster/bin/cxfs_info`
Command that normally invokes the client daemon:	`/etc/init.d/cxfs_client`
Failover file	`/etc/failover2.conf`
GRIO v2 administration	`/usr/sbin/grioadmin`
GRIO monitoring	`/usr/sbin/griomon`
GRIO v2 quality of service	`/usr/sbin/grioqos`
Hostname/address information	`/etc/hosts`
Log file:	`/var/log/cxfs_client`
Options file:	`/etc/cluster/config/cxfs_client.options`
XVM query	`/sbin/xvm`

Table A-2 Mac OS X Paths

Component	Path
CXFS client daemon:	`/usr/cluster/bin/cxfs_client`
Command that normally invokes the client daemon:	`/Library/StartupItems/cxfs/cxfs`
CXFS status:	`/usr/cluster/bin/cxfs_info`
Failover file	`/etc/failover2.conf`
GRIO v2 administration	`/usr/sbin/grioadmin`
GRIO monitoring	`/usr/sbin/griomon`
GRIO v2 quality of service	`/usr/sbin/grioqos`
Hostname/address information	`/etc/hosts`
Log file:	`/var/log/cxfs_client`
Options file:	`/usr/cluster/bin/cxfs_client.options`
XVM query	`/usr/cluster/bin/xvm`

Table A-3 Windows Paths

Component	Path
CXFS client service:	`%SystemRoot%\system32\cxfs_client.exe`
Command that normally invokes the client service:	See "Start/Stop the `cxfs_client` Service for Windows" on page 164
CXFS status:	`%ProgramFiles%\CXFS\cxfs_info.exe`
Failover file	`%ProgramFiles%\CXFS\failover2.conf`
GRIO v2 administration:	`%ProgramFiles%\CXFS\grioadmin.exe`
GRIO monitoring:	`%ProgramFiles%\CXFS\griomon.exe`
GRIO v2 quality of service:	`%ProgramFiles%\CXFS\grioqos.exe`
Hostname and address information:	`%SystemRoot%\system32\drivers\etc\hosts`
Log file:	`%ProgramFiles%\CXFS\log\cxfs_client.log`
Options file:	See "Modifying the CXFS Software for Windows" on page 168
XVM query:	(unsupported)

Appendix B

Filesystem and Logical Unit Specifications

Table B-1 on page 228 summarizes filesystem and logical unit specifications differences among the supported client-only platforms.

Table B-1 Filesystem and Logical Unit Specifications

Item	Linux x86_64	Linux ia64	Mac OS X	Windows
Maximum filesystem size	2^{64} bytes	2^{64} bytes	2^{64} bytes	2^{64} bytes
Maximum file size / offset	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes	2^{63}–1 bytes
Filesystem block size (in bytes)[1]	512, 1024, 2048, or 4096	512, 1024, 2048, 4096, 8192, or 16384	4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536
Physical block size (in bytes) supported by XVM	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536	512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536
Physical LUN limit for GPT-labeled disks [2]	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks	2^{63} device blocks
Maximum concatenated slices	65536	65536	65536	65536

1 If the filesystem is to be accessible by other platforms in a multiOS cluster, its block size must be supported on all platforms in the cluster.

2 Note the following about physical LUN limits for GPT-labeled disks:

- Physical LUNs with GPT labels are not constrained by XVM or CXFS to be smaller than the largest possible filesystem.
- Cluster nodes may constrain the LUN size to be smaller due to driver or other operating system constraints. A LUN used in the cluster may not be larger than the maximum size allowed by any node.
- Windows XP does not support LUNs greater than 2 TB in size. Filesystem corruption will occur if you attempt to write to the LUN above the 2-TB boundary. CXFS for Windows will not allow a filesystem to be mounted if any part of it resides on a LUN that is greater than 2–TB in size.

Mount Options Support

The table in this appendix list the mount options that are supported by CXFS, depending upon the server platform. Some of these mount options affect only server behavior and are ignored by client-only nodes.

The tables also list those options that are not supported, especially where that support varies from one platform to another. The `mount` commands supports many additional options, but these options may be silently ignored by the clients, or cause the mount to fail and should be avoided. For more information, see the `mount`(8) man page.

Note: The following are mandatory, internal CXFS mount options that cannot be modified and are set by `clconfd` and `cxfs_client`:

```
client_timeout
server_list
```

The table uses the following abbreviations:

Y = Yes, client checks for the option and sets flag/fields for the metadata server
N = No, client does not check for the option
S = Supported
n = Not supported (when set, the **Checked by Client** column does not apply)
D = Determined by the CXFS administration tools (not user-configurable)

Table C-1 Mount Options Support for Client-Only Platforms

Option	Checked by Supporting Clients	Linux	Mac OS X	Windows
agskip	N	S [1]	S	S
allocsize	Y	S	S	S
attr2	N	n	n	n
biosize [2]	Y	S	S	S
client_timeout	Y	D	D	D
dmi	Y	S	S	S
filestreams [3]	Y	S	S	S
gqnoenforce	Y	S	S	S
gquota	Y	S	S	S
grpid	Y	S	S	S
grpquota	Y	S	S	S
ibound	Y	S	S	S
inode64	Y	S	S	S

1 `agskip` is not supported on RHEL 4 and RHEL 5

2 On an ia64 Linux node with a page size of 64K, the `biosize` value must be at least `16`.

3 Do not use the `dmi` and `filestreams` options together. DMF is not able to arrange file extents on disk in a contiguous fashion when restoring offline files. This means that a DMF-managed filesystem most likely will not maintain the file layouts or performance characteristics normally associated with filesystems using the `filestreams` mount option.

Option	Checked by Supporting Clients	Linux	Mac OS X	Windows
largeio	Y	S	S	S
logbsize	Y	S	S	S
logbufs	Y	S	S	S
logdev	N	S	S	S
mrquota	Y	S	n	n
noalign	Y	S	n	S
noatime	Y	S	S	S
noattr2	N	n	n	n
noauto	N	n	n	n
nobarrier	N	S	S	S
nodev	N	S	S	S
nodiratime	N	S	n	n
noexec	Y	S	n	n
nolargeio	N	S	n	n
noquota	Y	S	S	S
nosuid	Y	S	S	S
osyncisdsync	Y	S	n	n
pqnoenforce	Y	S	S	S

Option	Checked by Supporting Clients	Linux	Mac OS X	Windows
pquota	Y	S	S	S
prjquota	Y	S	S	S
qnoenforce	Y	S	S	S
quota	Y	S	S	S
relatime	Y	n	n	n
ro	Y	S	S	S
rtdev	N	n	n	n
rw	N	S	S	S
server_list	Y	D	D	D
server_timeout	Y	D	D	D
strictatime	Y	n	n	n
sunit	Y	S	S	S
swalloc	Y	S	S	S
swidth	Y	S	S	S
uqnoenforce	Y	S	S	S
uquota	Y	S	S	S
usrquota	Y	S	S	S
wsync	Y	S	S	S

Error Messages

The following are commonly seen error messages:

Could Not Start CXFS Client Error Messages

The following error message indicates that the `cxfs_client` service has failed the license checks:

```
Could not start the CXFS Client service on Local Computer.

Error 10038: An operation was attempted on something that is not a socket.
```

You must install the license as appropriate. See the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

CMS Error Messages

The following messages may be logged by CMS.

`CMS excluded cells 0x`*XXX* `with incomplete connectivity`

Generated when CMS delivers a membership that excluded some **new** cells that had not established connections with enough cells yet to be admitted. 0x*XXX* is a bitmask of excluded cells.

`CMS calculation limited to last membership:configuration change incomplete on cells 0x`*XXX*

Generated when the leader is attempting to make a configuration change current (that is, actually use the change on all nodes), but some cells in the cluster have not yet received the configuration change staged (uploaded and ready to be made current). `0x`*XXX* is a bitmask of cells that do not yet have the change in their configuration. Changes make their way through the cluster asynchronously, so this situation is expected. It can take a few attempts by the CMS leader before all nodes have the change staged. As long as this situation resolves eventually, there is no problem.

```
CMS calculation limited to last membership:recovery incomplete
```

Generated when new members were disallowed due to recovery from the last cell failure that is still being processed.

Mount Messages

```
cxfs_client:op_failed ERROR: Mount failed for concat0
```

A filesystem mount has failed and will be retried.

Network Connectivity Messages

```
unable to join multicast group on interface
unable to create multicast socket
unable to allocate interface list
unable query interfaces
failed to configure any interfaces
unable to create multicast socket
unable to bind socket
```

Check the network configuration of the node, ensuring that the private network is working and the Windows node can at least reach the metadata server by using the `ping` command from a command shell.

Device Busy Message

You may see the following error message repeatedly on a node when you stop services on another node until the shutdown completes:

```
Nov  4 15:35:12 ray : Nov 04 15:35:12 cxfs_client:
cis_cms_exclude_cell ERROR: exclude cellset ffffffffffffff00 failed: Device busy
```

After the other node completes shutdown, the error will cease to be sent. However, if the error message continues to appear even after shutdown is complete, another problem may be present. In this case, contact your SGI support person.

Windows Messages

The following are common Windows CXFS messages.

`cis_driver_init() failed: could not open handle to driver`
`cis_driver_init() failed: could not close handle to CXFS driver`

The CXFS driver may not have successfully started. Check the system event log for errors.

`cis_generate_userid_map warning: could not open group file`

The `group` file could not be found.

Even with `passwd` and `group` warnings above, filesystem mounts should proceed; however, all users will be given `nobody` credentials and will be unable to view or modify files on the CXFS filesystems. For more information about these files, see "Log Files and Cluster Status for Windows" on page 106. Also see the log files on the server-capable administration node; for more information, see the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

`cis_generate_userid_map warning: could not open passwd file`

The `passwd` file could not be found.

`could not get location of passwd/group files`
`could not retreving fencing configuration file name from registry`
`error retrieving passwd filename`
`error retrieving group filename`
`error retrieving fencing filename`

The registry entries for the location of the `passwd`, `group`, or `fencing.conf` files may be missing, or the path provided on the command line to the CXFS Client service is badly formed. Reset these values by modifying the current installation as described in "Modifying the CXFS Software for Windows" on page 168.

`could not open passwd file`
`could not open group file`
`fencing configuration file not found`
: Check that the `passwd`, `group` and `fencing.conf` files are in the configured location and are accessible as described in "Checking Permissions on the Password and Group Files for Windows" on page 145.

`no valid users configured in passwd file`
: No users in the `passwd` file could be matched to users on the Windows node. All users will be treated as user `nobody` for the purpose of all access control checks.

`no valid groups configured in group file`
: No groups in the `group` file could be matched to groups on the Windows node. Attempts to display file permissions will most likely fail with the message `Unknown Group Errors`.

`op_failed ERROR: Mount failed for concat0`
: A filesystem mount has failed and will be retried.

`unable to create mount point`
`Configured drive letter may already be in use`
: Check that the configured drive letter is not already in use by a physical or mapped drive.

`Unix user is something other than a user on the NT domain/workgroup`
`Unix group is something other than a group on the NT domain/workgroup`
: This warning indicates that a username or groupname is not a valid user or group on the Windows node, which may be confusing when examining file permissions.

L2 System Controller for Linux Reset

This section discusses the following:

L2 System Controller Reset Configuration

You should use a separate L2 for each node to avoid unnecessary reboots. The L2 system controller and the required USB cables are optional equipment available for purchase.

Note: Configurations that use TTY ports require that you purchase serial cables.

You can use network reset if you have an L2 on a network. For details, see the information about `reset_comms` in the `cxfs_admin` chapter of *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

The L2 controller must not be on the primary CXFS private network. Ideally, the L2 controller should be on a different private network and must be reachable by all server-capable administration nodes in the cluster. A public network is not ideal for security reasons, but is acceptable. The number of network connections allowed depends upon the L2 version; contact your SGI support person for assistance.

SGI ia64 systems with an integrated L2 (such as a NUMAlink® 4 R-brick), Altix 3000 Bx2 systems, Altix 450 systems and Altix 4700 systems use the L2 over Ethernet. See Figure E-2.

In Altix 350, use IO10 and a *multiport serial adapter cable*, which is a device that provides four DB9 serial ports from a 36-pin connector; see Figure E-3.

Use the modem port on the L2 system controller as shown in Figure E-4. Use DB9 serial ports on an IX-brick on Altix 3000. Connect the serial cable to the modem port on one end and the serial port on the IX-brick (for example, serial port connector 0), as shown in Figure E-5.

Figure E-1 shows the L2 access via the Ethernet port on an A450.

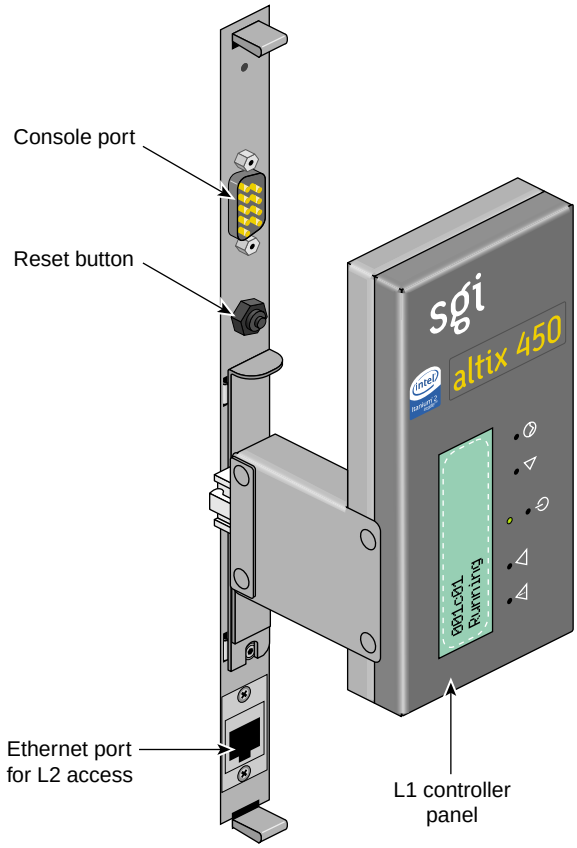

Figure E-1 L2 Access via the Ethernet Port on an A450

Figure E-6 shows connections for two machines with an L2 system controller. (This figure shows direct attached storage. Connections for other storage configurations will be the same.)

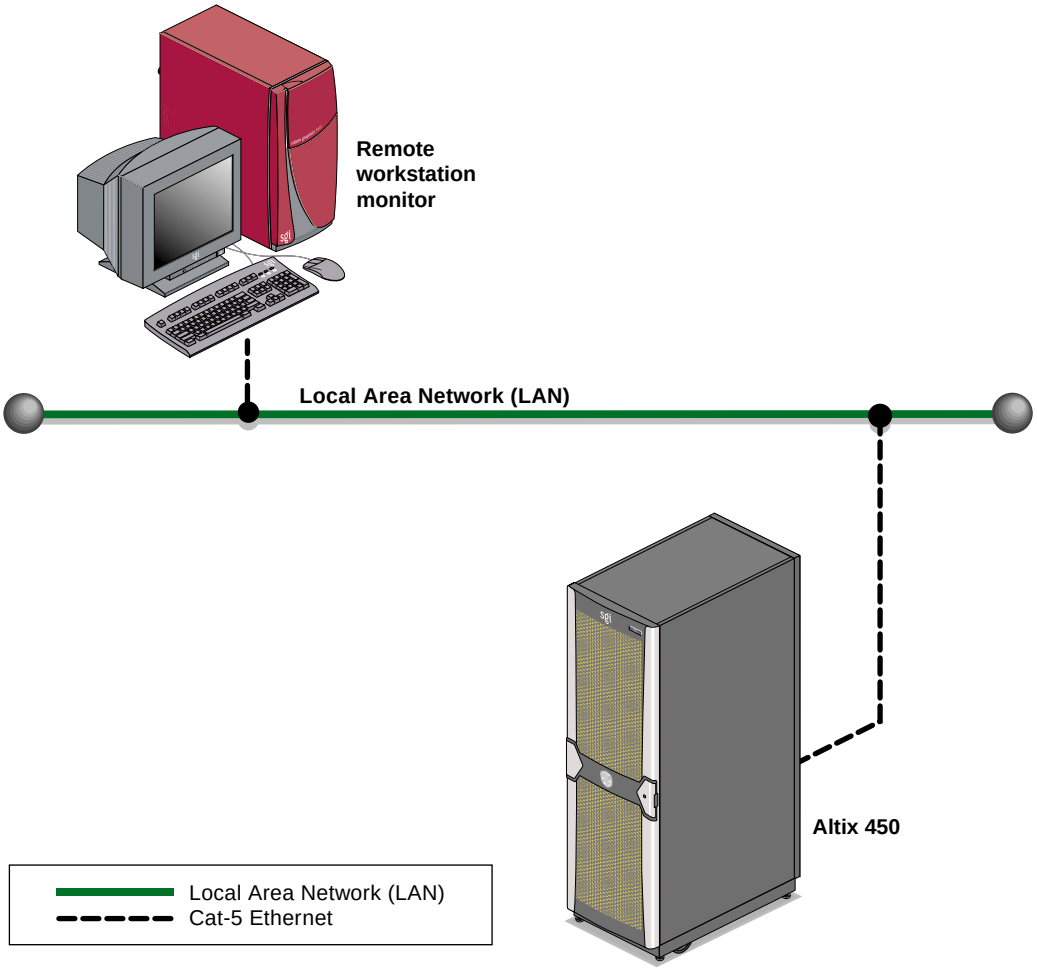

Figure E-2 SGI Altix 450 System Control Network

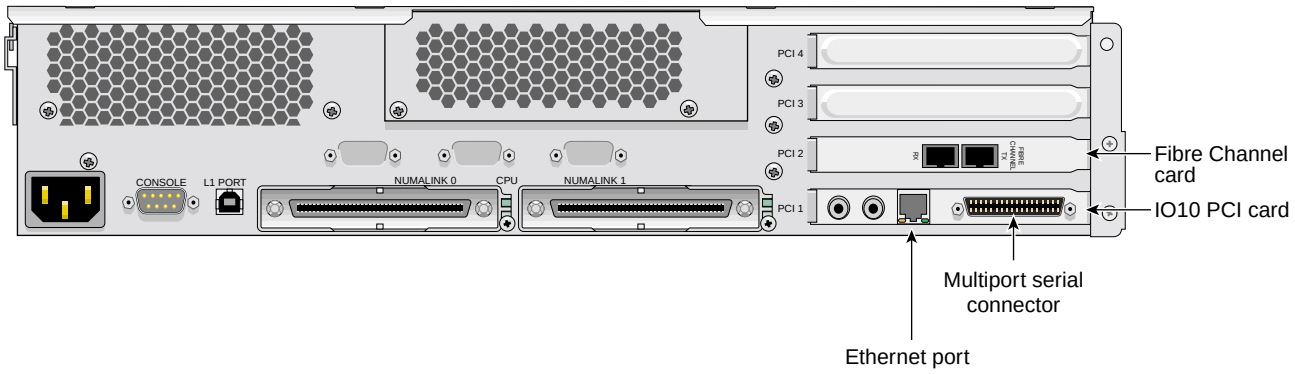

Figure E-3 Altix 350 Rear Panel

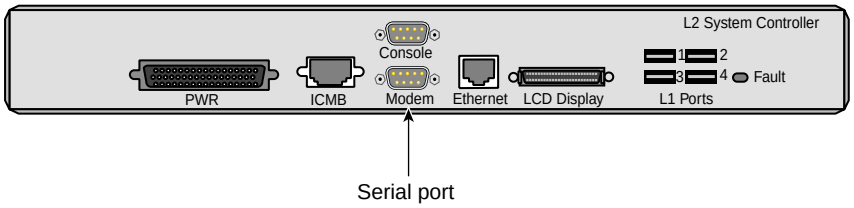

Figure E-4 L2 Rear Panel

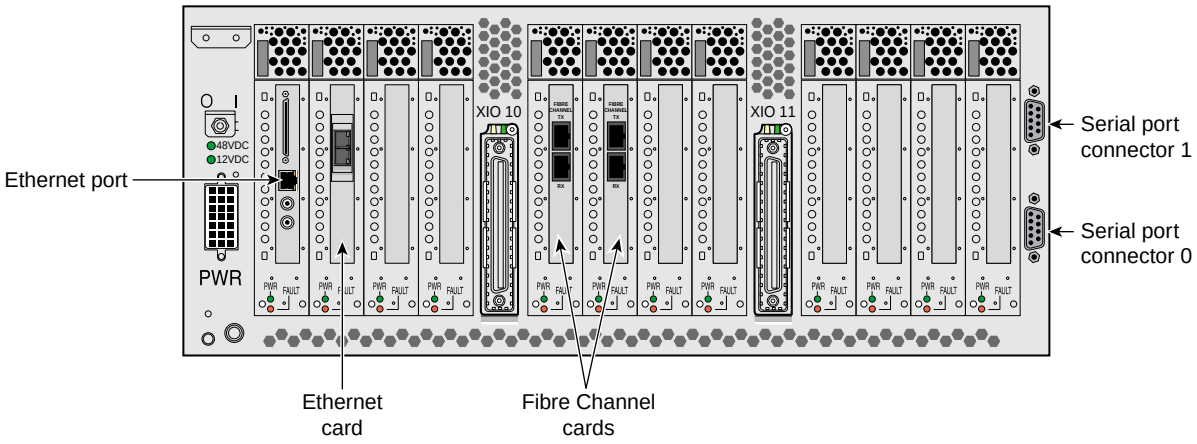

Figure E-5 IX-brick Rear Panel

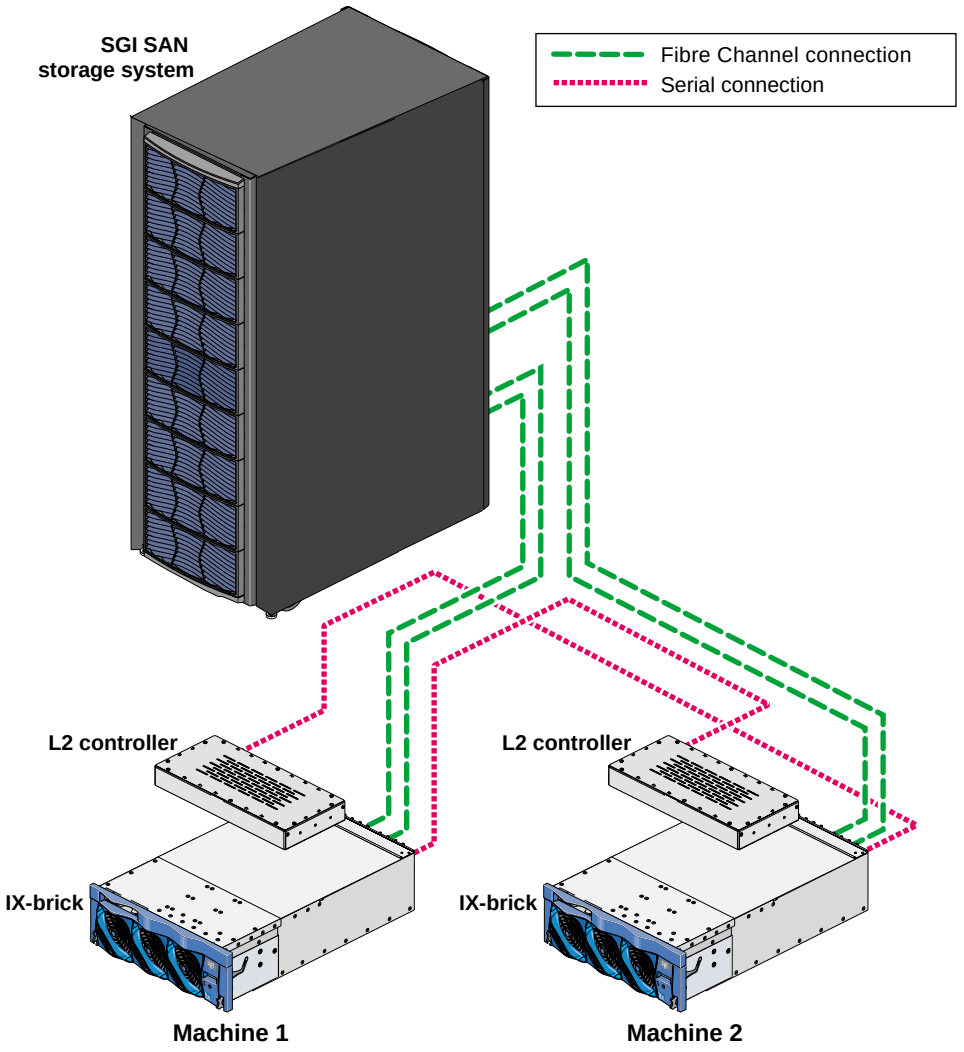

Figure E-6 Altix 3000 Connections

Testing Serial Connectivity for the L2 on Altix® 350 Systems

You can use the cu(1) command to test the serial reset lines if you have installed the uucp RPM.

The cu command requires that the device files be readable and writable by the user uucp. The command also requires the /var/lock directory be writable by group uucp.

Perform the following steps:

1. Change ownership of the serial devices so that they are in group uucp and owned by user uucp.

   Note: The ownership change may not be persistent across reboots.

   For example, suppose you have the following TTY devices on the IO10:

   ```
   server-admin# ls -l /dev/ttyIOC*
   crw-rw----  1 root uucp 204, 50 Sep 15 16:20 /dev/ttyIOC0
   crw-rw----  1 root uucp 204, 51 Sep 15 16:20 /dev/ttyIOC1
   crw-rw----  1 root uucp 204, 52 Sep 15 16:20 /dev/ttyIOC2
   crw-rw----  1 root uucp 204, 53 Sep 15 16:20 /dev/ttyIOC3
   ```

   To change ownership of them to uucp, you would enter the following:

   ```
   server-admin# chown uucp.uucp /dev/ttyIOC*
   ```

2. Determine if group uucp can write to the /var/lock directory and change permissions if necessary.

   For example, the following shows that group uucp cannot write to the directory:

   ```
   server-admin# ls -ld /var/lock
   drwxr-xr-t  5 root uucp 88 Sep 19 08:21 /var/lock
   ```

   The following adds write permission for group uucp:

   ```
   server-admin# chmod g+w /var/lock
   ```

3. Join the uucp group temporarily, if necessary, and use cu to test the line.

   For example:

   ```
   server-admin# newgrp uucp
   server-admin# cu -l /dev/ttyIOC0 -s 38400
   ```

For more information, see the cu(1) man page and the documentation that comes with the uucp RPM.

Summary of New Features from Previous Releases

This appendix contains a summary of the new features for each version of this guide.

CXFS MultiOS 2.0

Original publication (007-4507-001) supporting Solaris client-only nodes in a multiOS cluster with IRIX metadata servers.

CXFS MultiOS 2.1

The 007-4507–002 update contains the following:

- Support for Windows NT nodes in a CXFS multiOS cluster. Platform-specific information is grouped into separate chapters.
- Support for up to four JNI HBAs in each CXFS Solaris node.

Note: JNI supports a maximum of four JNI HBAs in operating environments with qualified Solaris platforms.

CXFS MultiOS 2.1.1

The 007-4507–003 update contains the following:

- References to using the latest software from the JNI website (http://www.jni.com/Drivers).
- Information about ensuring that appropriate software is installed on the IRIX nodes that are potential metadata servers.
- Clarifications to the use of I/O fencing and serial reset.
- Corrections to the procedure in the “Solaris Installation Overview” section and other editorial corrections.

CXFS MultiOS 2.2

The 007-4507–004 update contains the following:

- Support for Microsoft Windows 2000 nodes in a CXFS MultiOS cluster. This guide uses *Windows* to refer to both Microsoft Windows NT and Microsoft Windows 2000 systems.
- Support for SGI TP9100s. For additional details, see the release notes.
- A new section about configuring two HBAs for failover operation.
- Support for the JNI 5.1.1 and later driver on Solaris clients, which simplifies the installation steps.
- DMAPI support for all platforms.
- Removal of the Solaris limitation requiring more kernel threads.

CXFS MultiOS 2.3

The 007-4507–005 update contains the following:

- Updated Brocade Fibre Channel switch firmware levels.
- Filename corrections the chapters about FLEXlm licensing for Windows and modifying CXFS software on a Solaris system.

CXFS MultiOS 2.4

The 007-4507–006 update contains the following:

- Support for Sun Microsystems Solaris 9 and specific Sun Fire systems.
- Support for the JNI EZ Fibre release 2.2.1 or later.
- A cluster of as many as 32 nodes, of which as many as 16 can be CXFS administration nodes; the rest will be client-only nodes.
- Information about the **Node Function** field, which replaces node weight. For Solaris and Windows nodes, **Client-Only** is automatically selected for you. Similar fields are provided for the `cmgr` command. For more information, see the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

- Clarification that if the primary HBA path is at fault during the Windows boot up (for example, if the Fibre Channel cable is disconnected), no failover to the secondary HBA path will occur. This is a limitation of the QLogic driver.
- Reference to the availability of cluster information on Windows nodes.
- Information about enabling Brocade Fibre Channel switch ports.
- Additional information about functional limitations specific to Windows, and performance considerations, and access controls.

CXFS MultiOS 2.5

The 007-4507–007 update contains the following:

- Support for the IBM® AIX® platform, Linux on supported 32-bit platforms, SGI ProPack™ for Linux on Altix® servers.
- Support for a cluster of up to 48 nodes, 16 of which can be CXFS administration nodes; the rest must be client-only nodes.
- For Windows nodes, user identification with lightweight directory access protocol (LDAP).
- Support of forced unmount of filesystems on Windows nodes.
- Information about protecting data integrity if JNI Fibre Channel cables are disconnected or fail.
- Support for the SGI TP9500 RAID.
- Support for the QLogic 2342 host bus adapter.
- Information about new `cxfs-reprobe` scripts on AIX, IRIX, Linux, and Solaris nodes. These scripts are run by either `clconfd` or `cxfs_client` when they need to reprobe the Fibre Channel controllers. The administrator may modify these scripts if needed.
- Information about setting the `ntcp_nodelay` system tunable parameter in order to provide adequate performance on file deletes.
- Automatic detection of HBAs is provided for Linux, Solaris, and Windows nodes.

CXFS MultiOS 3.0

The 007-4507–008 update contains the following:

- Support for the Microsoft Windows XP client.

Note: The CXFS multiOS 3.0 release is the last release that will support the Microsoft Windows NT 4.0 platform. The 3.1 release will not include software for Windows NT 4.0.

- Clarifications to the terminology and installation information for Linux 32-bit clients.
- Information about Linux 64-bit clients running SGI ProPack for Linux on SGI Altix 3000 systems has been removed and will appear in the *CXFS 5 Administration Guide for SGI InfiniteStorage* that support CXFS 3.0 for SGI ProPack 2.3 for Linux.

CXFS MultiOS 3.1

The 007-4507–009 update contains the following:

- Support for the Apple Computer, Inc. Mac OS X operating system on client-only nodes.
- Support for a cluster of up to 64 nodes.
- Information about the SGI TP9300, SGI TP9300S, and SGI TP9500S.
- Information about setting the LUN discovery method for Solaris systems using the SGI TP9100 1-Gbit controller
- Additional AIX troubleshooting information.

CXFS MultiOS 3.2

The 007–4507–010 update contains the following:

- Support for Mac OS X 10.3.5 and Apple host bust adapters (HBAs).

Note: Mac OS X 10.2.*x* and the Astera HBA are not supported with the CXFS 3.2 release.

- Support for Red Hat Enterprise Linux 3. If you are running a Red Hat Enterprise Linux 3 kernel and you want to use quotas on a CXFS filesystem, you must install the quota package.
- Support for the Sun Fire V210 server as a multiOS client platform.
- A summary of the maximum filesystem size, file size, and block size for each platform.
- Information about the environment variables you must define in the `/etc/cluster/config/cxfs_client.options` file in order for the `/etc/cluster/config/cxfs-reprobe` script to appropriately probe all of the targets on the SCSI bus for the Linux platform on third-party hardware.
- Availability of the new `xvm_maxdmasz` attribute to the AIX `chdev` command, used to change the maximum XVM direct memory access (DMA) size to improve direct I/O performance.
- Information about ensuring proper hostname configuration for a Windows node.
- XVM volume names are limited to 31 characters and subvolumes are limited to 26 characters.
- Information about mount options.
- Updates to the procedure for installing the AMCC JNI HBA.
- Clarification that the AMCC JNI HBA that is provided by Sun Microsystems **does not function with CXFS** and cannot be configured to do so. You must purchase the JNI HBA directly from AMCC.

CXFS MultiOS 3.3

The 007–4507–011 update contains the following:

- Support for Microsoft Windows Server 2003.
- Support for AMD AMD64, Intel EM64T, and Intel Itanium 2 third-party Linux systems as client-only nodes.

- Information about guaranteed-rate I/O (GRIO) version 2 (v2).
- Information about XVM failover v2.
- Platform-specific information about FLEXlm licenses and troubleshooting has been separated out into the various platform-specific chapters.
- Information about the recognizing changes to the storage systems.
- System tunables information for Solaris and Windows.
- Information about the SANshare license and XVM failover v2 on AIX.
- Information about configuring HBA failover on Windows.
- New sections about verifying the cluster configuration, connectivity, and status.
- Removed references to `xvmprobe`. The functionality of `xvmprobe` has been replaced by the `xvm` command.

CXFS MultiOS 3.4

The 007–4507–012 update contains the following:

- Support for SUSE Linux Enterprise Server 9 (SLES9)
- Best practices for client-only nodes
- Mapping XVM volumes to storage targets on AIX and Linux
- Remote core dump on Mac OS X
- Installing the LSI Logic HBA

CXFS 4.0

The 007–4507–013 update contains the following:

- Support for the following:
  - Red Hat Enterprise Linux 4.

Note: On Red Hat Enterprise Linux 4 (RHEL4) x86 nodes, you must fully disable SELinux and redirect `core` dump files in order to avoid a stack overflow panic.

  - Mac OS X 10.4, including full ACL support.
  - Solaris 10.

  The following are not included in CXFS 4.0:

  - AIX 5.2
  - Red Hat Enterprise Linux 3
  - Mac OS X 10.3.9
  - Solaris 8

- Support for the `cxfs_admin` command
- Information about choosing the correct version of XVM failover for your cluster.
- If Norton Ghost is installed on a Windows node, CXFS cannot mount filesystems on the mount point driver letter.
- Information about using fast copying for large CXFS files
- A platform-independent overview of client-only installation process
- Server-side CXFS client license keys are now supported on server-capable nodes, allowing a client without a node-locked client-side license key to request a license key from the server. Server-side license keys are optional on IRIX metadata servers, but are required on SGI ProPack metadata servers. The licensing software is based on the FLEXlm product from Macrovision Corporation. See *CXFS 5 Administration Guide for SGI InfiniteStorage*.
- Information about configuring firewalls for CXFS use and membership being prevented by inappropriate firewall configuration
- Information about the maximum CXFS I/O request size for AIX
- Support for Apple PCI Express HBA.
- Support for QLogic HBA for the Solaris platform.

- Support for the CXFS `autopsy` and `fabric_dump` scripts on Mac OS X.

CXFS 4.1

The 007–4507–014 update contains the following:

- Support for SUSE Linux Enterprise Server 10 (SLES 10) client-only nodes

  Note: DMAPI is disabled by default on SLES 10 systems. If you want to mount filesystems on a SLES 10 client-only node with the `dmi` mount option, you must enable DMAPI.

- Support for SGI License Key (LK) software on SGI ProPack server-capable nodes.

  Server-side licensing is required on the following client-only nodes (to determine the Linux architecture type, use the `uname -i` command):

  – SGI ProPack 5

  – Red Hat Enterprise Linux (RHEL) 4 on x86_64

  – SLES 9 on x86_64

  – SLES 10 on x86_64 or ia64

  (For specific release levels, see the release notes.)

  Other nodes can use either server-side or client-side licensing. However, if one node within a cluster requires server-side licensing, all nodes must use server-side licensing. If no nodes in the cluster require server-side licensing, the nodes can continue to use existing client-side licensing.

  Note: Server-side licensing is preferred, and no new client-side licenses will be issued. Customers with support contracts can exchange their existing client-side licenses for new server-side licenses. A future release will not support client-side licensing. For more information, contact SGI customer support.

  For licensing details, see the release notes and the *CXFS 5 Administration Guide for SGI InfiniteStorage*.

- Support for changes in the Mac OS X device paths used by the `xvm` and `failover2.conf` files.
- A new chapter to support SGI Altix XE as a client-only node.
- Updates to the supported mount options tables.

CXFS 4.2

The 007–4507–015 update contains the following:

- Support for the following new platforms:
  - Mac OS X on the Intel platform
  - Windows 2003 x86_64 platform
- As of CXFS 4.2, all server-capable nodes running 4.2 and client-only nodes running 4.2 require server-side licensing. If **all** existing client-only nodes are running a prior supported release, they may continue to use client-side license as part of the rolling upgrade policy until they are upgraded to 4.2. All client-only nodes in the cluster must use the same licensing type — if any client-only node in the cluster is upgraded to 4.2 or if a new 4.2 client-only node is added, then all nodes must use server-side licensing. Customers with support contracts can exchange their existing client-side licenses for new server-side licenses. For more information, contact SGI customer support.
- Support for 4Gb PICx and PCIe HBA support on Windows nodes
- Support for GPT labels on the Mac OS X and Windows platforms
- Memory-mapped files flush time for Windows
- Mapping XVM volumes to storage targets on Windows
- XVM failover V2 on Windows
- Documentation for the support of XVM failover version 2 on Windows nodes (first supported in the CXFS 4.1.1 release).
- Clarifications about support for the following:
  - Real-time subvolumes
  - External logs

- Information about the `cmgr` command has been moved to an appendix. The preferred CXFS configuration tools are `cxfs_admin` and the CXFS graphical user interface (GUI). As of the CXFS 5.0 release, the `cmgr` command will not be supported or documented.
- Removal of support for the following:
  - AIX 5.2
  - SLES 9 SP3
  - SGI ProPack 4 SP 3
  - Solaris 9
  - Windows 2000 and Windows XP SP 1

CXFS 5.0

The 007–4507–016 version includes the following changes:

- Support for the following new platforms:
  - Mac OS X Leopard (10.5).
  - SGI ProPack 5 SP 4 (client-only) and SGI ProPack 5 SP 5 (server and client-only).
  - Windows:

    Windows Server 2003 SP2
    Windows Server SP2 x64
    Windows Vista
    Windows Vista x64
- The IRIX platform as a client-only node.
- Removed support for Linux i386 architecture.
- The new section “Mapping Physical Device Names to XVM Physvols.”

CXFS 5.2

The 007–4507–017 version includes the following changes:

- CXFS server-capable nodes must run SGI Foundation Software 1.

  SGI Foundation Software 1 is a new product from SGI consisting of technical support tools, utilities, and driver software that enable SGI's Linux systems to run reliably and consistently. SGI ProPack 6 is the next generation of SGI's suite of performance-optimization libraries and tools that accelerate applications on SGI's Linux systems. SGI ProPack 6 may be optionally installed on any CXFS node running SGI Foundation Software 1.

- Support for *edge serving*, in which CXFS client nodes can act as servers for NFS, Samba, CIFS, or any third-party network filesystem exporting files from a CXFS filesystem. However, there are no performance guarantees when using edge serving; for best performance, SGI still recommends that you use the active metadata server. If you require a high-performance solution, contact SGI Professional Services.

- Clarifications to the list of supported mount options for the Windows platform.

- Clarification that the physical LUN limit with GPT-labeled disks is 2 TB for IRIX 6.5.28 and IRIX 6.5.29 nodes.

CXFS 5.4

The 007–4507–018 version includes the following changes:

- Clarifications about the need to reboot a Linux node after enabling GRIO.

- Information about the fact that the `cxfs_client` software automatically detects the world wide port names (WWPNs) of any supported host bus adapters (HBAs) for Solaris nodes that are connected to a switch that is configured in the cluster database. (Introduced in CXFS 5.3.) See

- "Warning: DiskManager for Windows Vista and Windows 2008 Destroys Data".

- "Saving Application Crash Dumps for Windows Vista and Windows 2008".

CXFS 5.6

The 007–4507–019 version includes support for running the following on client-only nodes:

- SLES 11
- Windows Vista Service Pack 2
- Windows Server 2008 Service Pack 2

Some caveats and considerations that were formerly listed in the CXFS general release note have been incorporated into this guide.

CXFS 6.0

The 007–5619–001 guide supersedes *CXFS 5 Client-Only Guide for SGI InfiniteStorage* (007–4507–019). This new guide includes the following:

- New support for the following:
  - Mac OS X Snow Leopard 10.6.2 or later
  - RHEL 4 U3
  - SGI Foundation 2
  - SGI ProPack 7
  - Windows 7
- Removal of information about the AIX, IRIX, and Solaris client-only platforms

---

Note: AIX, IRIX, and Solaris clients are not supported in ISSP 2.0 and 2.*X* releases going forward. The AIX, IRIX, and Solaris clients are still fully supported in the CXFS 5.*X* series in ISSP 1.*X*.

CXFS 6.2

The 007–5619–002 guide includes the following:

- Availability of the `cxfs_admin` command on Mac OS X client-only nodes.
- Removal of the section “Windows Server 2008 Marks Newly Discovered Disks Offline”. Due to a fix in this release, you should no longer use DiskManager to mark the disks `Online`. Instead, CXFS™ now uses the `Offline` disk feature to prevent Windows from attempting to initialize and format newly discovered disks. (Attempting to mark a CXFS disk as `Online` will fail and a `permission denied` error message will appear.)

CXFS 6.4

The 007–5619–003 guide includes the following:

- Support for the Red Hat Enterprise Linux 5.6 (RHEL 5.6) client-only node.
- As of ISSP 2.3, XVM device names on Mac OS X have been changed from `/dev/[r]xvm*` to `/dev/[r]disk-xvm*` to follow the Mac OS X device naming convention. If you are upgrading from an earlier release, you must modify the contents of the `/etc/failover2.conf` file accordingly.
- Removal of the `large_xattr_action` parameter on Mac OS X nodes and addition of the new `large_resourcefork_xa_action` parameter, which specifies how files with large resource fork extended attributes (those larger than 64 KB) will be handled on a CXFS filesystem on a Mac OS X node.
- Addition of the new `AppleDouble` file format for resource fork attributes for Mac OS X nodes.
- Clarifications about setting system tunable parameters on Linux nodes and Mac OS X nodes and improved organization of the information for Windows nodes.

  For details about the available Linux parameters, see the appendix in the *CXFS 7 Administrator Guide for SGI InfiniteStorage*.

CXFS 6.6

The 007–5619–004 guide includes the following:

- Information about disabling token prefetch and range tokens

  Note: You should disable these features only at the direction of SGI Support.

- Support for all physical LUN block sizes up to 64 KB (512, 1024, 2048, 4096, 8192, 16384, 32768, or 65536 bytes) on Linux and Mac OS X platforms
- Support for Mac OS X Lion (10.7 or later)
- Support for ATTO host bus adapters (HBAs) for Mac OS X

CXFS 7.0

The 007–5619–005 guide includes the following:

- Support for new platform versions:
  - Mac OS X Mountain Lion (10.8.1 or later)
  - Windows 8
- Clarifications about issues with memory-mapped files:
- Information about L2 configuration for system reset of Linux client-only nodes is now available in an appendix.
- Information about preallocating space for directions when appropriate, because some Windows applications with a high ratio of metadata transfer or small I/O do not work efficiently on CXFS clients.
- XVM support for physical block sizes greater than 512 bytes on Windows.

CXFS 7.1

The 007–5619–006 guide includes the following:

- The `xvm.exe` command on Windows now provides persistent path naming for Fibre Channel RAID. You can continue to use existing `failover2.conf` files, but you can also choose to create a new file using the persistent paths.

  The section about configuring I/O fencing for Windows

- Support for the Windows HBAAPI library, which eliminates the need to manually configure an I/O `fencing.conf` file for Windows. This requires that you install the HBA driver.

- Clarifications and corrections

CXFS 7.2

The 007–5619–007 guide includes corrections and clarifications

CXFS 7.3

The 007–5619–008 contains the following changes:

- Revised instructions for the Linux client installation.

- The path to the `start` and `stop` scripts for CXFS has changed to `/usr/cluster/bin/cxfs`

- Added references to Mac OS X Mavericks (first supported in CXFS 7.2) and Mac OS X Yosemite (first supported in CXFS 7.3)

- Clarifications and corrections

CXFS 7.4

The 007–5619–009 contains the following changes:

- Support for the RHEL 7 and SLES 12 clients, including the use of the `systemctl`(1) command rather than the `chkconfig`(8) and `service`(8) commands.

- Modifying CXFS folder permissions on Windows.
- Clarifications and corrections.

Glossary

ACL

Access control list.

active metadata server

A server-capable administration node chosen from the list of potential metadata servers. There can be only one active metadata server for any one filesystem. See also *metadata*.

administration node

See *server-capable administration node*.

administrative stop

See *forced CXFS shutdown*.

advanced mode

The `cxfs_admin` complexity mode that provides a list of possible choices when using the `<TAB>` key, prompts for all possible fields, displays all attributes, and includes debugging information in output.

ARP

Address resolution protocol.

basic mode

The `cxfs_admin` complexity mode that only shows the common options and attributes in `show` output, provides a list of possible choices when using the `<TAB>` key, and uses prompting.

bandwidth

Maximum capacity for data transfer.

blacklisted

A node that is explicitly not permitted to be automatically configured into the cluster database.

BMC

Baseboard management controller.

cell ID

A number associated with a node that is allocated when a node is added into the cluster definition with the GUI or `cxfs_admin`. The first node in the cluster has cell ID of 0, and each subsequent node added gets the next available (incremental) cell ID. If a node is removed from the cluster definition, its cell ID becomes available. It is not the same thing as the *node ID*.

CLI

Underlying command-line interface commands used by the CXFS Manager graphical user interface (GUI).

client

In CXFS, a node other than the active metadata server that mounts a CXFS filesystem. A *server-capable administration node* can function as either an active metadata server or as a CXFS client, depending upon how it is configured and whether it is chosen to be the active metadata server. A *client-only node* always functions as a client.

client-only node

A node that is installed with the `cxfs_client.sw.base` software product; it does not run cluster administration daemons and is not capable of coordinating CXFS metadata. Any node can be client-only node. See also *server-capable administration node*.

cluster

A *cluster* is the set of systems (nodes) configured to work together as a single computing resource. A cluster is identified by a simple name and a cluster ID. A cluster running multiple operating systems is known as a *multiOS cluster*.

There is only one cluster that may be formed from a given pool of nodes.

Disks or logical units (LUNs) are assigned to clusters by recording the name of the cluster on the disk (or LUN). Thus, if any disk is accessible (via a SAN connection) from machines in multiple clusters, then those clusters must have unique names. When members of a cluster send messages to each other, they identify their cluster via the cluster ID. Cluster names must be unique.

Because of the above restrictions on cluster names and cluster IDs, and because cluster names and cluster IDs cannot be changed once the cluster is created (without deleting the cluster and recreating it), SGI advises that you choose unique names and cluster IDs for each of the clusters within your organization.

cluster administration daemons

The set of daemons on a server-capable administration node that provide the cluster infrastructure: `cad`, `cmond`, `fs2d`, `crsd`.

cluster administration tools

The CXFS graphical interface (GUI) and the `cxfs_admin` command-line tools that let you configure and administer a CXFS cluster, and other tools that let you monitor the state of the cluster.

cluster administrator

The person responsible for managing and maintaining a cluster.

cluster database

The database that contains configuration information about all nodes and the cluster. The database is managed by the cluster administration daemons.

cluster database membership

The group of server-capable administration nodes in the **pool** that are accessible to cluster administration daemons and therefore are able to receive cluster database updates; this may be a subset of the nodes defined in the pool. The cluster administration daemons manage the distribution of the cluster database (CDB) across the server-capable administration nodes in the pool. (Also known as *user-space membership* and *fs2d database membership*.)

cluster domain

The XVM concept in which a filesystem applies to the entire cluster, not just to the local node. See also *local domain*.

cluster ID

A unique number within your network in the range 1 through 255. The cluster ID is used by the operating system kernel to make sure that it does not accept cluster information from any other cluster that may be on the network. The kernel does not use the database for communication, so it requires the cluster ID in order to verify cluster communications. This information in the kernel cannot be changed after it has been initialized; therefore, you must not change a cluster ID after the cluster has been defined. Clusters IDs must be unique.

cluster mode

One of two methods of CXFS cluster operation, `Normal` or `Experimental`. In `Normal` mode, CXFS monitors and acts upon CXFS kernel heartbeat or cluster database heartbeat failure; in `Experimental` mode, CXFS ignores heartbeat failure. `Experimental` mode allows you to use the kernel debugger (which stops heartbeat) without causing node failures. You should only use `Experimental` mode during debugging with approval from SGI support.

complexity mode

The manner in which `cxfs_admin` operates. See *basic mode* and *advanced mode*.

control messages

Messages that the cluster software sends between the cluster nodes to request operations on or distribute information about cluster nodes. Control messages, CXFS kernel heartbeat messages, CXFS metadata, and cluster database heartbeat messages are sent through a node's network interfaces that have been attached to a private network.

cluster node

A node that is defined as part of the cluster. See also *node*.

control network

See *private network*.

CXFS

Clustered XFS, a clustered filesystem for high-performance computing environments.

CXFS client daemon

The daemon (`cxfs_client`) that controls CXFS services on a client-only node.

CXFS control daemon

The daemon (`clconfd`) that controls CXFS services on a server-capable administration node.

CXFS database

See *cluster database*.

CXFS kernel membership

The group of CXFS nodes that can share filesystems in the cluster, which may be a subset of the nodes defined in a cluster. During the boot process, a node applies for CXFS kernel membership. Once accepted, the node can share the filesystems of the cluster. (Also known as *kernel-space membership*.) CXFS kernel membership differs from *cluster database membership*.

CXFS services

The enabling / disabling of a node, which changes a flag in the cluster database. This disabling / enabling does not affect the daemons involved. The daemons that control CXFS services are `clconfd` on a server-capable administration node and `cxfs_client` on a client-only node.

CXFS services start

To enable a node, which changes a flag in the cluster database, by using an administrative task in the CXFS GUI or the `cxfs_admin enable` command.

CXFS services stop

To disable a node, which changes a flag in the cluster database, by using the CXFS GUI or the `cxfs_admin disable` command. See also *forced CXFS shutdown*.

CXFS shutdown

See *forced CXFS shutdown* and *shutdown*.

CXFS tiebreaker node

A node identified as a tiebreaker for CXFS to use in the process of computing CXFS kernel membership for the cluster, when exactly half the nodes in the cluster are up and can communicate with each other. There is no default CXFS tiebreaker. SGI recommends that the tiebreaker node be a client-only node.

database

See *cluster database*.

database membership

See *cluster database membership*.

details area

The portion of the GUI window that displays details about a selected component in the view area. See also *view area*.

domain

See *cluster domain* and *local domain*.

dynamic heartbeat monitoring

Starts monitoring CXFS kernel heartbeat only when an operation is pending. Once monitoring initiates, it monitors at 1-second intervals and declares a timeout after 5 consecutive missed seconds, just like *static heartbeat monitoring*.

easy client configuration

Using the `cxfs_admin` command and the `autoconf` object to specify new client-only nodes that are allowed to be automatically configured into the cluster database.

edge-serving

See *NFS edge-serving*.

fail policy hierarchy

See *fail policy*.

failure policy

The set of instructions that determine what happens to a failed node; the second instruction will be followed only if the first instruction fails; the third instruction will be followed only if the first and second fail. The available actions are: *fence*, *fencereset*, *reset*, and *shutdown*.

fence

The failure policy method that isolates a problem node so that it cannot access I/O devices, and therefore cannot corrupt data in the shared CXFS filesystem. I/O fencing can be applied to any node in the cluster (CXFS clients and metadata servers). The rest of the cluster can begin immediate recovery.

fencereset

The failure policy method that fences the node and then, if the node is successfully fenced, performs an asynchronous system reset; recovery begins without waiting for reset acknowledgment. If used, this fail policy method should be specified first. If the fencing action fails, the reset is not performed; therefore, `reset` alone is also highly recommended for all server-capable administration nodes (unless there is a single server-capable administration node in the cluster).

fencing recovery

The process of recovery from fencing, in which the affected node automatically withdraws from the CXFS kernel membership, unmounts all filesystems that are using an I/O path via fenced HBA(s), and then rejoins the cluster.

forced CXFS shutdown

The withdrawl of a node from the CXFS kernel membership, either due to the fact that the node has failed somehow or by issuing an `admin cxfs_stop` command. This disables filesystem and cluster volume access for the node. The node remains enabled in the cluster database. See also *CXFS services stop* and *shutdown*.

fs2d database membership

See *cluster database membership*.

gratuitous ARP

ARP that broadcasts the MAC address to IP address mappings on a specified interface.

GUI

Graphical user interface. The CXFS GUI lets you set up and administer CXFS filesystems and XVM logical volumes. It also provides icons representing status and structure.

GPT

GUID partition table

heartbeat messages

Messages that cluster software sends between the nodes that indicate a node is up and running. CXFS kernel heartbeat messages, cluster database heartbeat messages, CXFS metadata, and control messages are sent through the node's network interfaces that have been attached to a private network.

heartbeat timeout

If no CXFS kernel heartbeat or cluster database heartbeat is received from a node in this period of time, the node is considered to be dead. The heartbeat timeout value must be at least 5 seconds for proper CXFS operation.

I/O fencing

See *fence*.

IPMI

Intelligent Platform Management Interface.

ISSP

SGI InfiniteStorage Software Platform, the distribution method for CXFS software.

kernel-space membership

See *CXFS kernel membership*.

LAN

Local area network.

local domain

XVM concept in which a filesystem applies only to the local node, not to the cluster. See also *cluster domain*.

log configuration

A log configuration has two parts: a *log level* and a *log file*, both associated with a *log group*. The cluster administrator can customize the location and amount of log output, and can specify a log configuration for all nodes or for only one node. For example, the `crsd` log group can be configured to log detailed level-10 messages to the `crsd-nodeA` log only on the node `nodeA` and to write only minimal level-1 messages to the `crsd` log on all other nodes.

log file

A file containing notifications for a particular *log group*. A log file is part of the *log configuration* for a log group.

log group

A set of one or more CXFS processes that use the same log configuration. A log group usually corresponds to one daemon, such as `gcd`.

log level

A number controlling the number of log messages that CXFS will write into an associated log group's log file. A log level is part of the log configuration for a log group.

logical volume

A logical organization of disk storage in XVM that enables an administrator to combine underlying physical disk storage into a single unit. Logical volumes behave like standard disk partitions. A logical volume allows a filesystem or raw device to be larger than the size of a physical disk. Using logical volumes can also increase disk I/O performance because a volume can be striped across more than one disk. Logical volumes can also be used to mirror data on different disks. For more information, see the *XVM Volume Manager Administrator Guide*.

LUN

Logical unit. A logical disk provided by a RAID. A logical unit number (LUN) is a representation of disk space. In a RAID, the disks are not individually visible because they are behind the RAID controller. The RAID controller will divide up the total disk space into multiple LUNs. The operating system sees a LUN as a hard disk. A LUN is what XVM uses as its physical volume (*physvol*). For more information, see the *XVM Volume Manager Administrator Guide*.

membership

See *cluster database membership* and *CXFS kernel membership*.

membership version

A number associated with a node's cell ID that indicates the number of times the CXFS kernel membership has changed since a node joined the membership.

metadata

Information that describes a file, such as the file's name, size, location, and permissions.

metadata server

The server-capable administration node that coordinates the updating of metadata on behalf of all nodes in a cluster. There can be multiple potential metadata servers, but only one is chosen to be the active metadata server for any one filesystem.

metadata server recovery

The process by which the metadata server moves from one node to another due to an interruption in CXFS services on the first node. See also *recovery*.

multiOS cluster

A cluster that is running multiple operating systems, such Linux and Windows.

multiport serial adapter cable

A device that provides four DB9 serial ports from a 36-pin connector.

NFS edge-serving

A configuration in which CXFS client nodes can export data with NFS.

node

A *node* is an operating system (OS) image, usually an individual computer. (This is different from the NUMA definition for a brick/blade on the end of a NUMAlink cable.)

A given node can be a member of only one pool and only one cluster. See also *client-only node*, *server-capable administration node*, and *standby node*.

node ID

An integer in the range 1 through 32767 that is unique among the nodes defined in the pool. You must not change the node ID number after the node has been defined. It differs from *cell ID*.

node membership

The list of nodes that are active (have CXFS kernel membership) in a cluster.

notification command

The command used to notify the cluster administrator of changes or failures in the cluster and nodes. The command must exist on every node in the cluster.

owner host

A system that can control a node remotely, such as power-cycling the node. At run time, the owner host must be defined as a node in the pool.

owner TTY name

The device file name of the terminal port (TTY) on the *owner host* to which the system controller is connected. The other end of the cable connects to the node with the system controller port, so the node can be controlled remotely by the owner host.

peer-to-disk

A model of data access in which the shared files are treated as local files by all of the hosts in the cluster. Each host can read and write the disks at near-local disk speeds; the data passes directly from the disks to the host requesting the I/O, without

passing through a data server or over a LAN. For the data path, each host is a peer on the SAN; each can have equally fast direct data paths to the shared disks.

physvol

Physical volume. A disk that has been labeled for use by XVM. For more information, see the *XVM Volume Manager Administrator Guide*.

pool

The set of nodes from which a particular cluster may be formed. Only one cluster may be configured from a given pool, and it need not contain all of the available nodes. (Other pools may exist, but each is disjoint from the other. They share no node or cluster definitions.)

A pool is formed when you connect to a given node and define that node in the cluster database using the CXFS GUI. You can then add other nodes to the pool by defining them while still connected to the first node, or to any other node that is already in the pool. (If you were to connect to another node and then define it, you would be creating a second pool).

port password

The password for the system controller port, usually set once in firmware or by setting jumper wires. (This is not the same as the node's `root` password.)

potential metadata server

A server-capable administration node that is listed in the metadata server list when defining a filesystem; only one node in the list will be chosen as the active metadata server.

private network

A network that is dedicated to CXFS kernel heartbeat messages, cluster database heartbeat messages, CXFS metadata, and control messages. The private network is accessible by administrators but not by users. Also known as *control network*.

quorum

The number of nodes required to form a cluster, which differs according to membership:

- For CXFS kernel membership:
  - A majority (**>50%**) of the server-capable administration nodes in the cluster are required to **form** an initial membership
  - Half (**50%**) of the server-capable administration nodes in the cluster are required to **maintain** an existing membership
- For cluster database membership, **50%** of the **nodes in the pool** are required to form and maintain a cluster.

quorum master

The node that is chosen to propagate the cluster database to the other server-capable administration nodes in the pool.

RAID

Redundant array of independent disks.

recovery

The process by which a node is removed from the CXFS kernel membership due to an interruption in CXFS services. It is during this process that the remaining nodes in the CXFS kernel membership resolve their state for cluster resources owned or shared with the removed node. See also *metadata server recovery*.

relocation

The process by which the metadata server moves from one node to another due to an administrative action; other services on the first node are not interrupted.

reset

The failure policy method that performs a system reset via the system controller.

SAN

Storage area network. A high-speed, scalable network of servers and storage devices that provides storage resource consolidation, enhanced data access, and centralized storage management.

server-capable administration node

A node that is installed with the `cluster_admin` product and is also capable of coordinating CXFS metadata.

server-side licensing

Licensing that uses license keys on the CXFS server-capable administration nodes; it does not require node-locked license keys on CXFS client-only nodes. The license keys are node-locked to each server-capable administration node and specify the number and size of client-only nodes that may join the cluster membership. All nodes require server-side licensing.

shutdown

The fail policy that tells the other nodes in the cluster to wait before reforming the CXFS kernel membership. The surviving cluster delays the beginning of recovery to allow the node time to complete the shutdown. See also *forced CXFS shutdown*.

split cluster

A situation in which cluster membership divides into two clusters due to an event (such as a network partition or an unresponsive server-capable administration node) and the lack of reset or CXFS tiebreaker capability. This results in multiple clusters, each claiming ownership of the same filesystems, which can result in filesystem data corruption. Also known as *split-brain syndrome*.

snooping

A security breach involving illicit viewing.

split-brain syndrome

See *split cluster*.

spoofing

A security breach in which one machine on the network masquerades as another.

standby node

A server-capable administration node that is configured as a potential metadata server for a given filesystem, but does not currently run any applications that will use that filesystem.

static heartbeat monitoring

Monitors CXFS kernel heartbeat constantly at 1-second intervals and declares a timeout after 5 consecutive missed seconds (default). See also *dynamic heartbeat monitoring*.

storage area network

See *SAN*.

system controller port

A port sitting on a node that provides a way to power-cycle the node remotely. Enabling or disabling a system controller port in the cluster database tells CXFS whether it can perform operations on the system controller port.

system log file

Log files in which system messages are stored.

tiebreaker node

See *CXFS tiebreaker node*.

transaction rates

I/O per second.

user-space membership

See *cluster database membership*.

view area

The portion of the GUI window that displays components graphically. See also *details area*.

VLAN

Virtual local area network.

whitelisted

A node that is explicitly allowed to be automatically configured into the cluster database.

XFS

A filesystem implementation type for the Linux operating system. It defines the format that is used to store data on disks managed by the filesystem.

Index

D

E

F

G

M

N

O

P

Q

T

U

V

W

X

Y